



MÒDUL DE CHATBOTS PER IATECH

Informe de Progrés I
Daniel García Nilo - 1633613

Índex de continguts

1. Nivell de seguiment de la planificació prevista i ajustaments	3
1.1. Seguiment i planificació de l'anàlisi.....	3
1.2. Seguiment i planificació del PoC.....	5
2. Canvis introduïts en els objectius fixats i/o en la metodologia de treball prevista	7
2.1. Canvis a l'anàlisi del modul de Chatbots.....	7
2.2. PoC basat en el chatbot de gestió d'incidències.....	8
2.3. Revisió dels requisits.....	9
2.3.1. Requisits funcionals.....	9
2.3.2. Requisits no funcionals.....	10
3. Metodologia del projecte.....	11
4. Progrés i evidències del treball realitzat	14
4.1. Introducció general del bloc.....	14
4.2. Anàlisi general i arquitectura del sistema de chatbots.....	14
4.2.1. Comprensió dels sistemes d'Analítica de Negoci i del proposit d'un chatbot	14
4.2.2. Diagrames de flux del comportament d'un chatbot base.....	16
4.2.3. Taules d'anàlisi i transicions	19
4.2.4. Modelatge de la base de dades.....	22
4.3. PoC: Chatbot de gestió d'incidències.....	26
4.3.1. Funcionament del PoC i els seus promts.....	26
4.3.2. Funcionament del flux conversacional del PoC.....	27
4.3.3. Evidències del funcionament real	31
4.3.4. Conclusions visuals i tècniques del bloc.....	35
5. Bibliografia	36

1. Nivell de seguiment de la planificació prevista i ajustaments

La planificació del projecte contempla dues línies de treball principals: l'anàlisi funcional i tècnica del mòdul de chatbots, i el desenvolupament del Proof of Concept (PoC) centrat en el chatbot de gestió d'incidències. L'anàlisi va iniciar-se formalment al **sprint 25.04**, mentre que el desenvolupament del PoC va començar al **sprint 25.05**. El punt final d'aquest projecte serà al finalitzar el sprint 25.08, per tant a les gràfiques i sprint 25.09 per posar com a referència a la finalització del sprint 25.08.

1.1. Seguiment i planificació de l'anàlisi

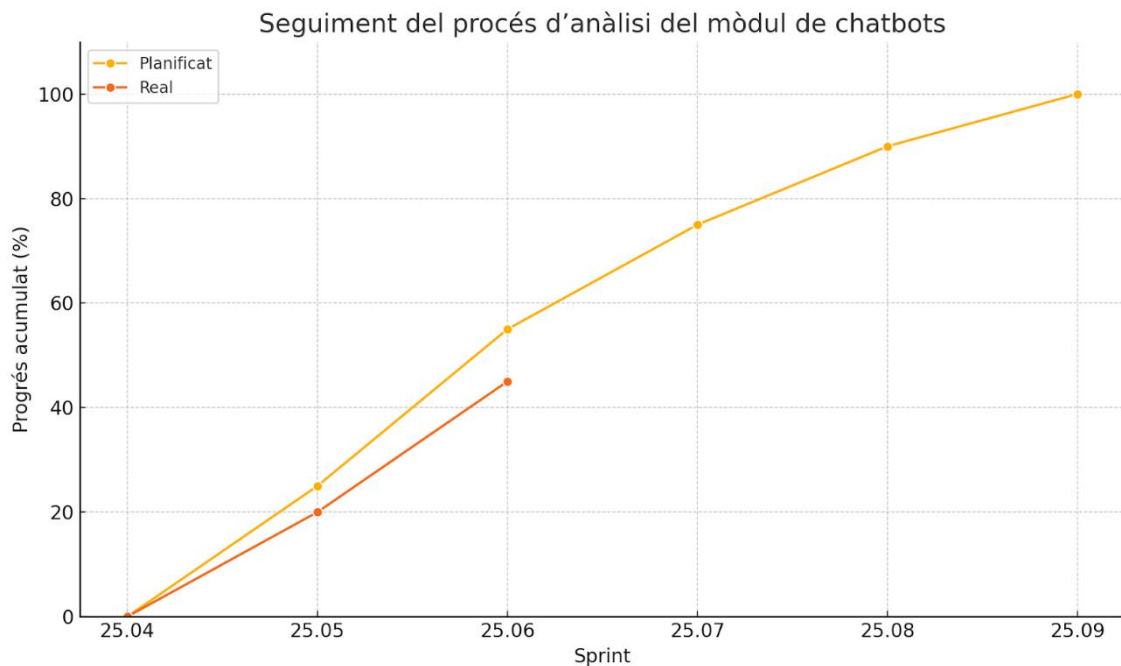
En relació amb la planificació inicial presentada al començament del TFG, s'ha produït un ajustament important en l'ordre i enfocament del treball. originalment, es preveia abordar des de l'inici el disseny i l'anàlisi dels diferents chatbots específics per a IA Tech. No obstant això, durant les primeres fases del projecte s'ha considerat més adient aprofundir en l'anàlisi conceptual i tècnic d'un chatbot base, per tal de garantir una base sòlida i reutilitzable que permeti, en fases posteriors, desplegar nous chatbots amb molta més eficiència i coherència arquitectònica.

El treball d'anàlisi del mòdul de chatbots ha avançat de manera significativa durant aquesta primera fase del projecte, amb un enfocament metòdic i profund orientat a garantir una base arquitectònica sòlida i escalable. En lloc de centrar-se des del principi en els chatbots específics de IA Tech, s'ha optat per analitzar amb detall el funcionament d'un **chatbot base genèric**, amb l'objectiu de definir una estructura reutilitzable i coherent que serveixi com a patró per a futurs desenvolupaments.

Aquest anàlisi ha inclòs l'elaboració de diversos **diagrames** i esquemes conceptuals que representen amb precisió el comportament intern del sistema. S'han identificat i classificat els **diferents tipus d'estats** (inicials, de classificació, de sol·licitud d'informació, de resolució, de generació de ticket, entre d'altres) i s'han definit clarament les seves **condicions de transició**. Aquest treball ha permès establir un **diagrama d'estats complet**, que descriu el cicle de vida d'una consulta des del seu inici fins a la resolució o escalat, incloent els casos en què s'integren elements com documents adjunts o validacions manuals per part de l'usuari.

Tasca prevista	Descripció inicial	Estat actual
Anàlisi dels chatbots específics de IA Tech	Definir funcionalitats i fluxos individualitzats per a cada bot	Endarrerit
Anàlisi de requisits globals	Recol·lecció i definició dels requisits funcionals i no funcionals	Completat
Anàlisi d'un chatbot estàndard reutilitzable	Establir un model base de comportament, fluxos i arquitectura general	En procés
Identificació d'estats i transicions	Anàlisi exhaustiva del comportament intern del chatbot (control de flux)	Completat
Modelització del flux conversacional	Representació visual del fil d'interaccions i dels cicles conversacionals	Completat
Estudi de la integració amb Analítica de Negoci	Identificació del procés d'enviament de consultes, recepció de protocols, etc.	Completat
Disseny de l'estructura de base de dades	Primer esbós d'una BD comuna per a tots els chatbots de IA Tech	En procés

La següent figura mostra el seguiment del procés d'anàlisi del mòdul de chatbots des del seu inici al sprint 25.04. S'hi reflecteix la comparativa entre el progrés planificat i el real en cada sprint, evidenciant una lleu desviació a la baixa que es considera assumible dins de la naturalesa exploratòria d'aquesta fase.



Tot i l'alt nivell d'activitat i concreció assolit fins ara, l'anàlisi funcional i tècnic del mòdul de chatbots encara requereix completar diversos aspectes clau. A continuació, es detallen les principals tasques d'anàlisi pendents, que s'han identificat com a necessàries per tancar la fase analítica amb garanties. Aquestes tasques, com que formen part d'un procés creat des de zero, poden evolucionar i adaptar-se lleugerament segons les necessitats que es vagin detectant durant el desenvolupament.

Tasca pendent	Descripció	Estat
Finalització del disseny de la base de dades	Definir esquema final, taules relacionades, camps específics per contextos, historial, etc.	Planificat
Model de governança del flux conversacional	Definir normes i mecanismes per controlar l'evolució d'una conversa, possibles errors i bifurcacions	Planificat
Disseny dels mecanismes de persistència d'estats	Establir com i quan guardar l'estat d'una conversa i com recuperar-lo si cal	Planificat
Estudi de mecanismes d'escalat entre bots	Establir condicions i interfícies per transferir consultes entre diferents bots	Planificat
Validació funcional amb escenaris teòrics	Simular casos típics de consulta per verificar la robustesa del model d'anàlisi	Planificat
Redacció formal de la documentació tècnica	Elaborar documentació completa de l'estructura d'anàlisi, fluxos i decisions preses	Planificat

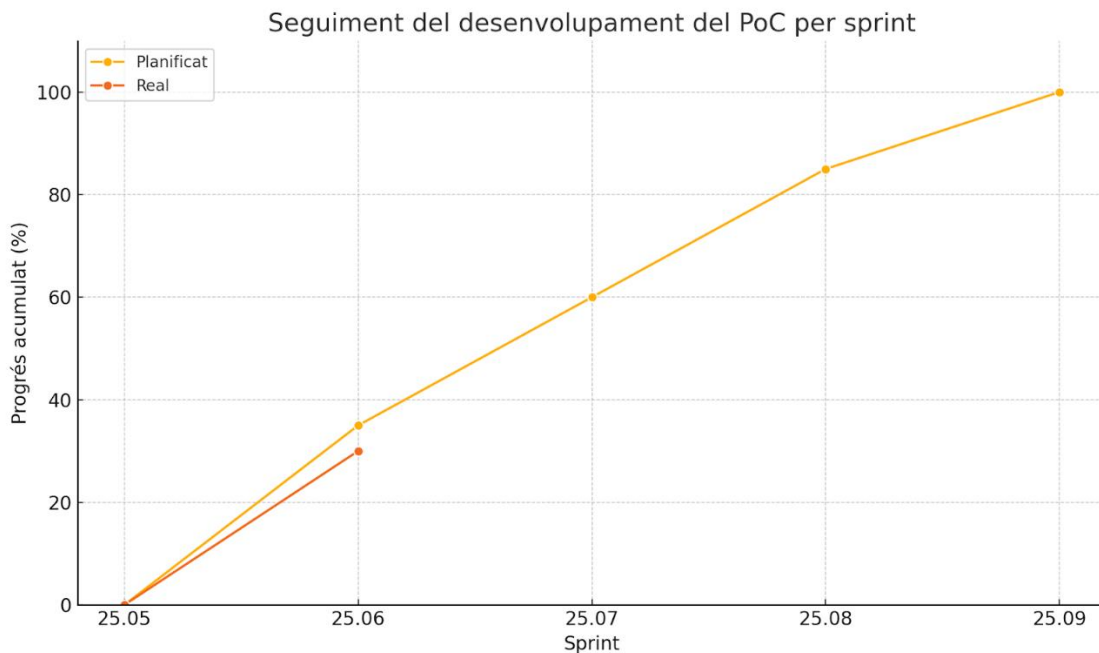
1.2. Seguiment i planificació del PoC

En paral·lel, per validar tècnicament l'enfocament, s'ha fet la primera part d'una **prova de concepte (PoC)** amb un dels casos concrets: el **chatbot de gestió d'incidències**, que integra-se amb el sistema JIRA per crear tiquets de suport tècnic de manera automàtica d'aquest chatbot ja s'ha completat un **anàlisi tècnic detallat del funcionament intern** que comentarem en profunditat en aquest informe.

Fins al moment, s'han desenvolupat i provat les següents funcionalitats:

Funcionalitat implementada	Descripció	Estat
Anàlisi i disseny del flux	Disseny detallat del comportament del chatbot, estats, transicions i estructura del flux	Completat
Inicialització del chatbot	Gestió de la sessió, generació de l'identificador únic i inicialització de la memòria temporal	Completat
Validació i control d'entrada	Comprovació de l'existència d'una conversa prèvia o creació d'una nova sessió	Completat
Gestió d'adjunts	Detecció de fitxers adjunts i generació de descripcions contextuais amb IA	Pràcticament complet
Detecció d'incidència	Ús de <code>detectar_incidencia_prompt</code> per confirmar si el missatge és una incidència real	Completat
Resposta inicial amb IA generativa	Generació de resposta automàtica a través del prompt <code>microsoft_prompt</code>	Completat

Tal com s'observa, en el moment actual (mitjan del sprint 25.06), **el progrés real se situa entorn del 30%, lleugerament per sota del 35% planificat**. Aquesta desviació és moderada i es considera assumible dins de la dinàmica del projecte.



Finalment, s'està començant a dissenyar l'**estructura de base de dades comuna** per als futurs chatbots de IA Tech, però aquest **PoC no l'utilitza**, ja que opera de manera autònoma mitjançant un sistema de memòria temporal i gestió directa de dades en temps real, pensat exclusivament per validar el flux funcional.

A continuació es presenten les funcionalitats pendents que es preveuen implementar en els propers sprints per completar el desenvolupament del PoC. Aquestes funcionalitats corresponen a la part més avançada del flux, centrada en la classificació, l'aplicació de protocols, la generació de tiquets i la validació final per part de l'usuari.

Tot i que el nombre de tasques pot semblar similar al de la primera fase, cal destacar que aquestes són significativament més **complexes a nivell funcional i tècnic**, ja que impliquen la integració amb diversos serveis externs (com el model de classificació AN o el backend de JIRA), així com la generació dinàmica de prompts amb lògica condicionada i interacció activa amb l'usuari per completar els passos del flux. Per aquest motiu, la seva execució requerirà més temps i detall tècnic que les tasques inicials.

Tasques pendents del PoC

Funcionalitat pendent	Descripció	Estat
Classificació d'incidències	Crida al model de classificació AN per determinar el component corresponent	Planificat
Obtenció de protocols	Lectura i processament de la Matriu de Categoritzacions per generar les opcions de resposta	Planificat
Classify prompt	Generació del prompt dinàmic i consulta a la IA per classificar la incidència dins d'un protocol concret	Planificat
Sol·licitud d'informació addicional	Ús del <code>info_prompt</code> per demanar més detalls a l'usuari segons el protocol assignat	Planificat
Execució d'accions automàtiques	Crida a la IA amb <code>action_prompt</code> per indicar a l'usuari els passos operatius a seguir	Planificat
Confirmació de la descripció JIRA	Presentació a l'usuari del resum generat per la IA abans de crear el tiquet	Planificat
Generació de ticket JIRA	Creació del tiquet amb les dades recopilades i enllaç de seguiment	Planificat
Validació final amb l'usuari	Confirmació per part de l'usuari abans de tancar el flux o escalar la incidència	Planificat

2. Canvis introduïts en els objectius fixats i/o en la metodologia de treball prevista

2.1. Canvis a l'anàlisi del modul de Chatbots

Pel que fa als **objectius**, la planificació original preveia definir i desenvolupar des de les primeres fases diversos **chatbots específics per a IA Tech**, cadascun orientat a una àrea funcional concreta. No obstant això, un cop iniciada la fase d'anàlisi, es va identificar la necessitat de construir primer un **model de chatbot base o estàndard**, capaç de servir com a estructura comuna i reutilitzable per a futurs bots. Aquest nou enfocament ha permès centrar esforços en la definició d'una arquitectura transversal, amb una gestió clara de contextos, estats, tipus d'interacció i integració amb fonts externes. Com a conseqüència, la definició dels chatbots específics s'ha ajornat per a una fase posterior, un cop el model base estigui validat.

Aquest canvi d'enfocament ha estat motivat per la voluntat de maximitzar la **reutilització de components i reduir la complexitat tècnica** futura. En lloc de dissenyar diversos bots amb estructures diferents, s'ha optat per establir un **marc de treball unificat** que defineixi com s'estructura un chatbot, com es relaciona amb la IA, com gestiona els estats interns i com es comunica amb sistemes externs com l'Analítica de Negoci o JIRA. Aquest model base permetrà aplicar patrons comuns i garantir una escalabilitat més eficient a mesura que es despleguin nous bots.

Anàlisi de la decisió: estudiar un chatbot base reutilitzable en lloc dels bots específics

Aspecte	Avantatges de centrar-se en un chatbot base	Inconvenients o limitacions
Escalabilitat i manteniment	Permet definir una arquitectura comuna aplicable a múltiples bots, simplificant futures implementacions	Endarrerirà la definició dels bots específics de IA Tech
Coherència arquitectònica	Garanteix que tots els bots comparteixin estructura i patrons tècnics coherents	Pot limitar la flexibilitat si algun bot requereix funcionalitats molt singulars
Robustesa tècnica	Possibilita detectar i resoldre problemes generals abans d'escapar-los als desenvolupaments específics	Requereix una major inversió inicial en anàlisi i disseny
Estandardització dels fluxos	Defineix una base de fluxos, estats i prompts reutilitzables en diferents casos d'ús	Pot demorar la personalització en fases inicials
Validació mitjançant PoC	El PoC del chatbot d'incidències valida pràcticament l'eficàcia del model general	El PoC no cobreix totes les particularitats dels altres chatbots

Aquesta decisió no implica una reducció de l'abast del projecte, sinó una **optimització de la seqüència de treball**: primer es valida el model genèric mitjançant una prova de concepte (PoC), i després es desenvoluparan els bots específics aprofitant tota la infraestructura comuna.

D'aquesta manera, es millora la **consistència arquitectònica**, es **reduïx la duplictat de codi i esforços**, i es garanteix una millor mantenibilitat del sistema global.

A nivell acadèmic, aquest ajustament també aporta valor, ja que el treball incorpora una **fase d'anàlisi més profunda**, amb propostes tècniques i metodològiques que poden ser útils per a altres contextos o projectes similars.

2.2. PoC basat en el chatbot de gestió d'incidències

Pel que fa al desenvolupament del **Proof of Concept (PoC)**, s'ha definit que aquest es realitzarà sobre el **chatbot de gestió d'incidències**, un dels casos més representatius i funcionals dins de l'ecosistema de IA Tech. Aquest chatbot és capaç d'identificar si una consulta és realment una incidència, i en cas afirmatiu, iniciar un flux de resolució automatitzat. Un cop detectada la incidència, el sistema consulta un **mòdul d'Analítica de Negoci**, que retorna un **component i un protocol** associat al tipus d'incidència. Aquest protocol s'utilitza per **filtrar les descripcions rellevants** dins de la **Matriu de Categoritzacions**, permetent així definir les accions, preguntes o missatges que ha de seguir el bot en cada cas.

A diferència del model final que s'implementarà per als bots futurs, aquest PoC **no farà ús del mòdul centralitzat de decisió**, també conegut com el **"cervell" o sistema d'orquestració intel·ligent**, que s'està definint dins de l'anàlisi general. En comptes d'això, el PoC opera mitjançant un **flux seqüencial predefinit**, on el chatbot segueix un **ordre de passos fix** activats per diferents **prompts** específics.

Aquests prompts són passats directament a un model de **IA Generativa**, que gestiona el diàleg en cada etapa del procés: primer es detecta si es tracta d'una incidència, després es proporciona una resposta si es pot resoldre automàticament, o bé es demana **més informació a l'usuari**. En cas que el protocol així ho indiqui, el chatbot pot **proposar una acció** que l'usuari hauria d'executar o, com a últim recurs, **crear automàticament un tiquet de suport** a través de JIRA.

Funcionalitat clau	Responsable principal	Descripció
Validació i inici de sessió	Sistema intern	S'inicialitza el flux, es crea l'identificador i la sessió de conversa amb memòria temporal.
Detecció d'incidències	IA Generativa (prompt específic)	Es determina si la consulta pot ser gestionada com una incidència.
Classificació amb Analítica de Negoci	Mòdul AN extern	El sistema envia la consulta al mòdul de classificació, que retorna un component i un protocol.
Consulta a la Matriu de Categoritzacions	Backend Matriu	Es recuperen accions i descripcions associades al protocol assignat.
Classificació final (IA)	IA Generativa (<code>classify_prompt</code>)	Es defineix quina descripció encaixa millor amb la incidència detectada.
Sol·licitud d'informació addicional	IA Generativa (<code>info_prompt</code>)	El bot pot demanar aclariments segons el que estableixi el protocol seleccionat.
Proposta d'accions específiques	IA Generativa (<code>action_prompt</code>)	El sistema suggereix accions que l'usuari pot executar per intentar resoldre la incidència.
Creació de tiquet JIRA	Backend JIRA + IA Generativa (<code>jira_prompt</code>)	Si no es pot resoldre, es genera un tiquet automàtic amb la informació recollida.
Validació de descripció amb l'usuari	IA Generativa (<code>jira_done</code>)	Es mostra al client un resum abans de crear el tiquet, per validar o modificar la descripció.

2.3. Revisió dels requisits

La definició d'aquest PoC ha comportat l'aparició de **nous requisits** que no estaven contemplats en la planificació inicial. Aquests requisits afecten tant el comportament funcional del sistema com les seves característiques no funcionals. El fet de treballar amb IA Generativa, d'integrar-se amb el sistema JIRA, i de gestionar un flux dinàmic i multietapa amb diferents fonts de coneixement (com la Matriu de Categoritzacions) ha evidenciat la necessitat de garantir aspectes com l'agilitat de resposta, la traçabilitat de la conversa, la gestió de sessions i la modularitat dels components. A més, s'ha posat de manifest la importància de tenir un sistema tolerant a errors i capaça de recuperar el context davant de situacions inesperades.

Aquests nous requisits s'han tingut en compte dins del disseny i implementació del PoC, amb la voluntat de validar no només el flux funcional sinó també la viabilitat tècnica i l'adequació als escenaris reals de suport tècnic.

2.3.1. Requisits funcionals

- El sistema ha de poder rebre i processar missatges textuais (i eventualment multimèdia) provinents dels usuaris.
- Cal generar un identificador únic per a cada conversa i recuperar el context associat si existeix.
- El sistema ha de determinar, mitjançant un prompt específic, si la consulta correspon a una incidència tècnica o no. **NOU**
- El sistema ha d'enviar la descripció al mòdul d'Analítica de Negoci per obtenir el component i protocol corresponents. **NOU**
- Un cop obtingut el protocol, s'han d'obtenir les possibles accions, missatges i preguntes definides per a aquest cas en la Matriu de Categoritzacions. **NOU**
- El sistema ha de crear prompts personalitzats amb les opcions de classificació o accions, segons el cas, integrant informació contextual. **NOU**
- La resposta del chatbot ha d'utilitzar models de llenguatge com ChatGPT per guiar l'usuari segons la classificació obtinguda.
- Si el protocol ho exigeix, el sistema ha de demanar informació específica a l'usuari abans de continuar. **NOU**
- El sistema ha de poder oferir a l'usuari una acció que pugui intentar per resoldre el problema abans d'escalar-lo. **NOU**
- El sistema ha de generar un resum automàtic per al tiquet JIRA i demanar confirmació o modificació a l'usuari abans de continuar **NOU**
- En cas necessari, s'ha de generar un tiquet de suport amb tota la informació estructurada (descripció, fitxers, historial...). **NOU**

- *El chatbot ha de notificar la resolució de la incidència o l'enllaç al seguiment del tiquet creat, segons correspongui.*

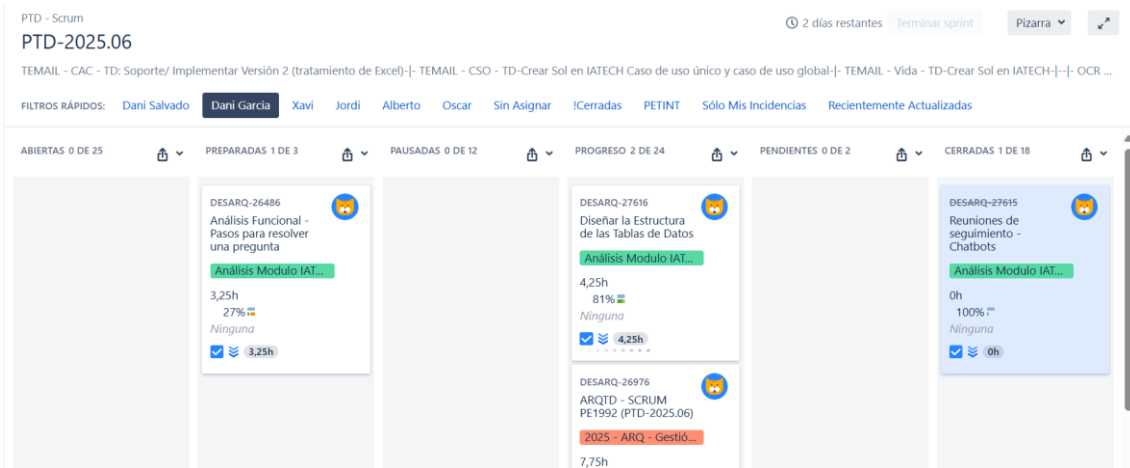
2.3.2. Requisits no funcionals

- *El sistema ha de ser escalable per permetre la incorporació de nous chatbots sense necessitat de redissenyar tota l'arquitectura.*
- *El sistema ha de disposar de mecanismes de seguretat per controlar l'accés a les dades i protegir la informació sensible.*
- *El sistema ha de complir amb les normatives de protecció de dades i garantir la privadesa dels usuaris.*
- *El sistema ha de garantir que les respostes generades siguin precises i comprensibles, evitant informació redundant o incorrecta.*
- *El sistema ha de ser modular per facilitar la mantenibilitat i l'actualització de components sense afectar el sistema global.*
- *El temps de resposta ha de ser prou curt per assegurar una experiència d'usuari fluida.* **NOU**
- *El sistema ha de gestionar entrades errònies o situacions imprevistes sense trencar la lògica de la conversa.* **NOU**
- *Cal que el sistema sigui capaç de mantenir el context actiu durant tota la sessió per garantir coherència en la interacció.* **NOU**
- *El sistema ha de garantir la integració fluida amb serveis com JIRA, models d'Analítica de Negoci o sistemes de gestió documental.* **NOU**
- *Cal disposar d'entorns de test i producció ben diferenciats per evitar interferències i validar canvis amb seguretat.* **NOU**

3. Metodología del proyecto

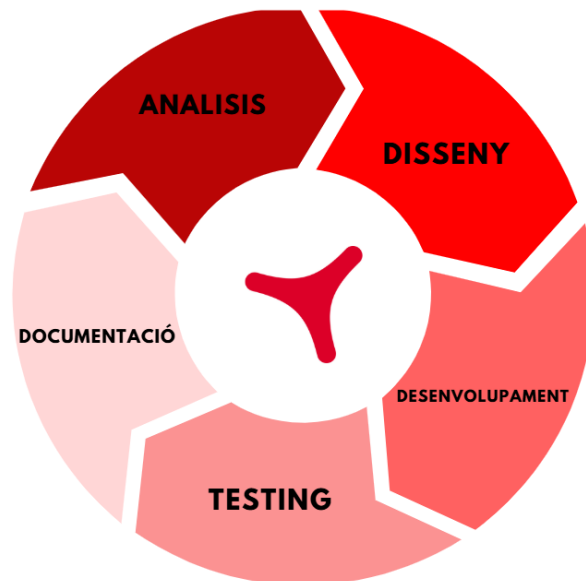
En el desenvolupament del projecte s'utilitza la metodologia **ScrumBan**, una combinació híbrida entre Scrum i Kanban que permet gestionar projectes amb estructures iteratives però també amb la flexibilitat necessària per adaptar-se a imprevistos o descobriments durant el procés. Aquesta metodologia és especialment útil en contextos de recerca aplicada o desenvolupament exploratori com el d'aquest projecte, en què les especificacions evolucionen i s'ajusten contínuament.

En concret, el treball es divideix en **sprints quinzenals** (25.04, 25.05, etc.), i per a cada sprint es defineixen un conjunt de tasques específiques agrupades per àrees (anàlisi, desenvolupament, proves, documentació...). Aquestes tasques es gestionen mitjançant un **tauler visual de seguiment**, amb columnes clàssiques de Oberta, Preparada, Pausada, En Progrés, Pendent i Tancada, i etiquetes que permeten identificar l'estat de maduresa de cada funcionalitat. Això ha ajudat a tenir un control constant de l'evolució del projecte i facilita l'adaptació dinàmica a canvis o prioritats.



A més de la gestió visual i iterativa, s'han seguit de manera clara les **fases típiques d'un cicle de desenvolupament àgil a cada iteració d'esprints**:

- **Anàlisi:** Amb una definició oberta i en constant evolució, orientada a crear una base reutilitzable per múltiples chatbots.
- **Disseny:** estructuració dels fluxos conversacionals, prompts, i arquitectura tècnica del sistema.
- **Desenvolupament incremental:** Implementació del PoC sobre el chatbot d'incidències, afegint funcionalitats en cada sprint.
- **Documentació iterativa:** Generació contínua de documentació tècnica i diagrames que acompanya cada fase i facilita la traçabilitat i validació del sistema.
- **Testing integrat dins del sprint:** fase de proves sobre el backend, utilitzant l'eina **WCF Test Client** per validar les primeres funcionalitats desenvolupades (inicialització del chatbot, detecció d'incidències, generació de respostes amb IA...). Aquest enfocament permet validar cada component tan bon punt es desenvolupa, seguint els principis de les metodologies àgils i facilitant l'evolució contínua del sistema.



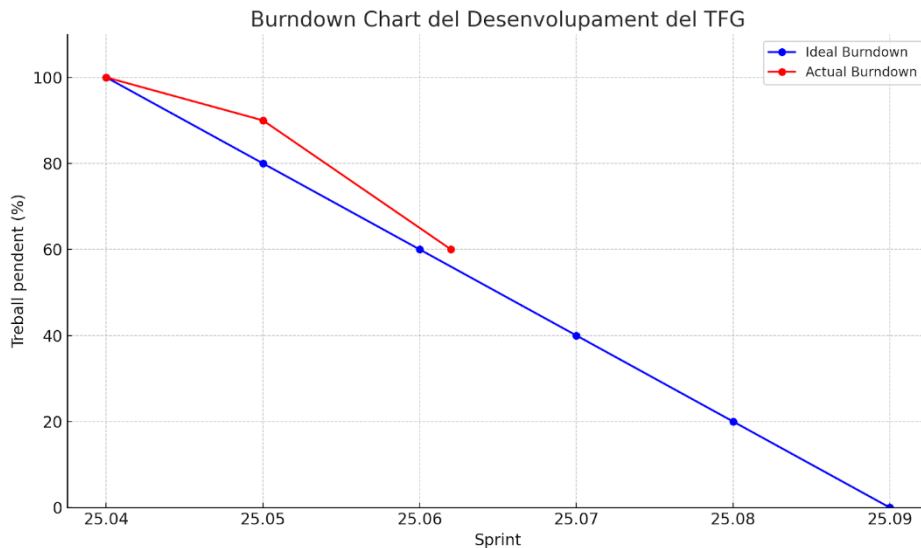
En el moment de lliurar aquest primer informe, el projecte es troba a **mitjan Sprint 25.06**, i s'ha assolit aproximadament un **35% del desenvolupament total estimat**, incloent tant el progrés en l'anàlisi com la primera fase de desenvolupament funcional del Proof of Concept.

Pel que fa al **compliment de la planificació inicial**, s'han assolit gran part de les fites previstes, si bé s'ha produït una lleugera desviació en la velocitat esperada, especialment durant les primeres iteracions. Aquest desfasament ha estat degut, sobretot, a la **complexitat afegida de l'anàlisi conceptual**, que ha requerit més temps del previst per definir l'arquitectura base, els estats del chatbot i les integracions amb sistemes externs com la IA Generativa, l'Analítica de Negoci i JIRA.

Per tal d'adaptar-se a aquests ajustaments, s'ha incorporat un **nou sprint addicional (25.09)** que marca el tram final del projecte. La previsió és que el TFG quedi **conclòs a l'inici del Sprint 25.09**, i, segons el calendari actual, hi ha temps suficient per completar també la **primera part d'aquest sprint**, destinada a tancar les funcionalitats pendents del PoC i validar-ne el comportament.

Malgrat aquesta extensió de la planificació, el projecte manté un **ritme estable i orientat a resultats**, tal com es reflecteix en la burndown chart inclosa en aquest informe. Tal com s'ha mostrat a l'apartat 1, el seguiment del projecte s'ha dividit en dues parts diferenciades: d'una banda, el desenvolupament del Proof of Concept (PoC) del chatbot d'incidències, i de l'altra, el procés d'anàlisi tècnica i funcional per establir una base comuna per a futurs chatbots. Ambdues parts han estat representades per gràfiques específiques de desviació respecte a la planificació inicial.

Tanmateix, si enfoquem el projecte des de la perspectiva de la metodologia **Scrumban**, la qual combina la planificació iterativa de Scrum amb la flexibilitat de Kanban, és convenient presentar una **burndown chart unificada** que reculli el **progrés global del TFG**. Aquesta gràfica reflecteix el descens acumulat del treball pendent al llarg dels sprints planificats. S'observa una evolució estable però lleugerament per sota del ritme ideal, finalitzant el sprint 25.06 amb aproximadament el **40% del treball completat**, enfront del 60% que s'havia previst inicialment.



La metodologia ScrumBan s'ha mostrat útil i eficient, permetent fer **revisions setmanals internes** del progrés, replanificar quan ha calgut, i mantenir un enfocament flexible però organitzat.

- *L'aplicació de la metodologia ScrumBan ha permès una adaptació constant del projecte a mesura que avançava, mantenint una visió clara del flux de treball i prioritzant la lliurabilitat contínua. Aquesta flexibilitat ha estat clau per ajustar l'enfocament inicial, replantejant els objectius de desenvolupament per centrar-se primer en una arquitectura de chatbot reutilitzable i sòlida.*
- *El projecte s'ha estructurat en sprints iteratius, amb fases definides d'anàlisi, disseny, desenvolupament, testing i documentació. Cada funcionalitat implementada ha estat validada dins del propi sprint mitjançant eines com WCF Test Client, garantint la qualitat i funcionalitat del sistema des del primer moment.*
- *Malgrat una lleu desviació temporal, el progrés acumulat és coherent amb la planificació general. S'han assolit fites importants tant en l'anàlisi com en la implementació del PoC, i es disposa d'una base sòlida per continuar amb el desenvolupament de les funcionalitats restants. Es preveu que, amb l'inici del sprint 25.07, es pugui recuperar el ritme estimat i avançar cap a la finalització del projecte sense comprometre'n l'abast ni la qualitat.*

4. Progrés i evidències del treball realitzat

4.1. Introducció general del bloc

Aquest apartat té com a objectiu mostrar, de forma organitzada i visual, totes les evidències tècniques i gràfiques que sustenten el progrés real del projecte. A través de diagrames, taules, esquemes, proves i captures del sistema, es busca no només confirmar que s'han assolit els objectius descrits a la planificació, sinó també facilitar la comprensió del funcionament intern del sistema, especialment per part de persones que no estiguin familiaritzades amb l'arquitectura dels chatbots o els sistemes d'IA.

Per tal d'oferir una visió estructurada i coherent, aquest bloc es divideix en dos grans blocs de contingut:

- *Anàlisi del sistema de chatbots i arquitectura general: on es documenta el treball realitzat per entendre i definir els fluxos conversacionals, els estats d'un chatbot base, la gestió de prompts, i el disseny d'una arquitectura transversal reutilitzable. Aquesta secció inclou també el model de dades proposat per als futurs bots de IA Tech.*
- *Desenvolupament i validació del PoC del chatbot d'incidències: es presenta la prova de concepte implementada per validar l'arquitectura anterior, centrada en un chatbot capaç de detectar incidències, gestionar respostes amb IA generativa i, si cal, obrir automàticament un tiquet a través de JIRA. Inclou captures de funcionament real, descripció del flux tècnic, diagrames de seqüència i proves realitzades sobre el backend.*

Aquest enfocament combinat entre teoria i pràctica permet evidenciar amb claredat la solidesa de l'anàlisi i la viabilitat de la implementació, deixant el projecte ben posicionat per avançar cap a la seva fase final.

4.2. Anàlisi general i arquitectura del sistema de chatbots

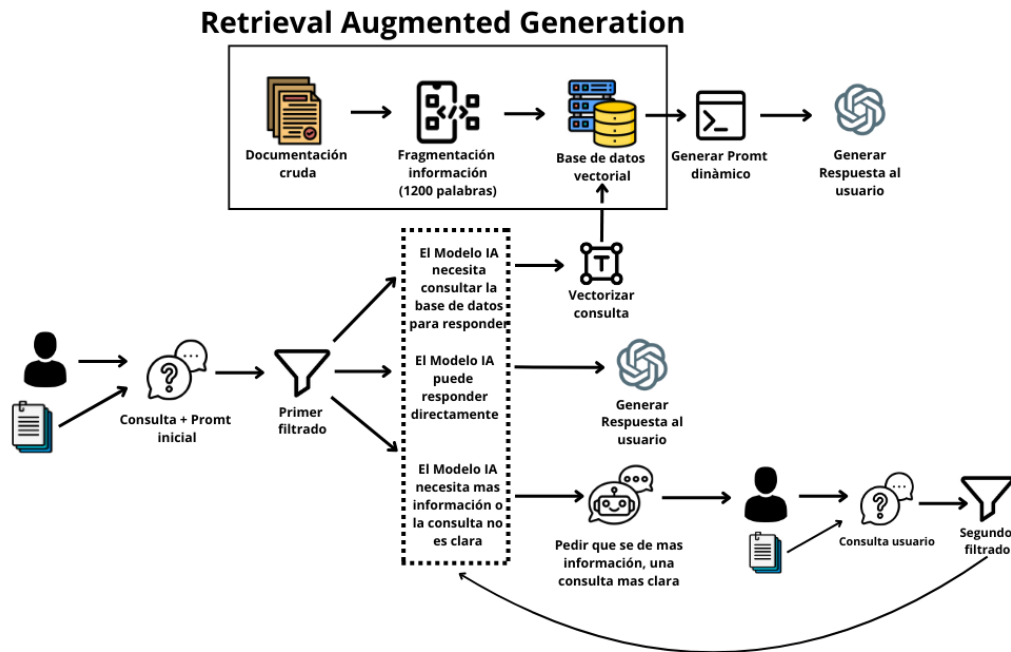
4.2.1. Comprensió dels sistemes d'Anàlisi de Negoci i del propòsit d'un chatbot

Durant les primeres setmanes del projecte es va dur a terme una reunió clau amb l'equip d'**Anàlisi de Negoci (AN)** de IA Tech. Aquesta trobada va permetre entendre amb profunditat com entenen ells el concepte de "chatbot" i quin és el seu ús actual dins de la seva arquitectura. A partir d'aquesta sessió es va obtenir un coneixement tècnic i funcional fonamental, que ha guiat tota la fase d'anàlisi posterior.

En primer lloc, es va introduir la figura del **Geppeto**, el nom amb què es fa referència als assistents intel·ligents capaços de recuperar informació rellevant de grans volums de documentació estructurada. AN va exposar la necessitat de millorar la **recuperació ràpida, precisa i contextualitzada d'informació**, especialment en entorns empresarials on l'usuari sovint ha de navegar entre múltiples documents, normatives i bases de dades. El bot actua com un agent intel·ligent que automatitza aquesta cerca i retorna respostes clares, sense que el client hagi d'endinsar-se manualment en grans blocs de contingut.

En aquest context, es va explicar com els **chatbots actuals d'Anàlisi de Negoci** utilitzen un sistema RAG, on es treballa sobre una **base de dades vectorial**, on la informació es fragmenta en blocs de 1.200 paraules aproximadament. Aquestes unitats es carreguen en un sistema d'embeddings semàntics, que permeten fer consultes més eficients i precises. Quan un usuari

envia una pregunta, el sistema avalua si pot respondre amb un prompt directe. Si no és el cas, el bot activa un **mecanisme de refinament**, amb possibles fases múltiples, per demanar més informació o aclariments. Finalment, un cop establert el context, es genera un **prompt dinàmic** que es passa al model de llenguatge, i aquest retorna una resposta acurada.



Tot aquest flux va permetre identificar que el model de chatbot que es planteja al TFG hauria de seguir un **procés estructurat i escalable**, amb anàlisi semàntic, classificació, refinament, generació i validació de respostes. Això ha inspirat el disseny del **chatbot base transversal** que s'ha anat construint durant la fase d'anàlisi.

A partir d'aquestes reflexions, també es va obrir un debat tècnic sobre l'ús de la tècnica coneguda com **RAG (Retrieval-Augmented Generation)**. Aquesta metodologia consisteix a enriquir el context d'un model de llenguatge amb informació rellevant recuperada de bases de dades, normalment vectorials. Si bé es va considerar que el **model base de chatbot** que s'està definint hauria de tenir **capacitat per integrar RAG en escenaris futurs**, es va decidir que el **PoC actual no incorporaria aquesta funcionalitat**. La decisió es justifica per diversos motius:

- El PoC se centra en validar un **flux seqüencial de prompts**, dissenyat per comprovar si el bot pot identificar una incidència, resoldre-la amb IA o escalar-la a través d'un ticket JIRA.
- No es treballa amb una base de coneixement externa ni amb volums alts de documentació per recuperar, per la qual cosa **la incorporació del RAG no és necessària**.
- Es vol mantenir la complexitat tècnica controlada, posant el focus en validar el **comportament del flux funcional** i no encara en l'optimització de la cerca de coneixement.

4.2.2. Diagrames de flux del comportament d'un chatbot base

Per tal d'abordar de manera estructurada el comportament intern d'un chatbot conversacional, s'ha seguit un procés d'anàlisi progressiu que parteix d'una perspectiva externa o de "caixa negra", i que ha derivat en una modelització detallada del flux intern dels passos que componen la conversa. Aquest procés ha estat fonamental per poder identificar els elements clau que intervenen en la interacció entre un usuari i un assistent virtual, així com per dissenyar una arquitectura flexible i reutilitzable.

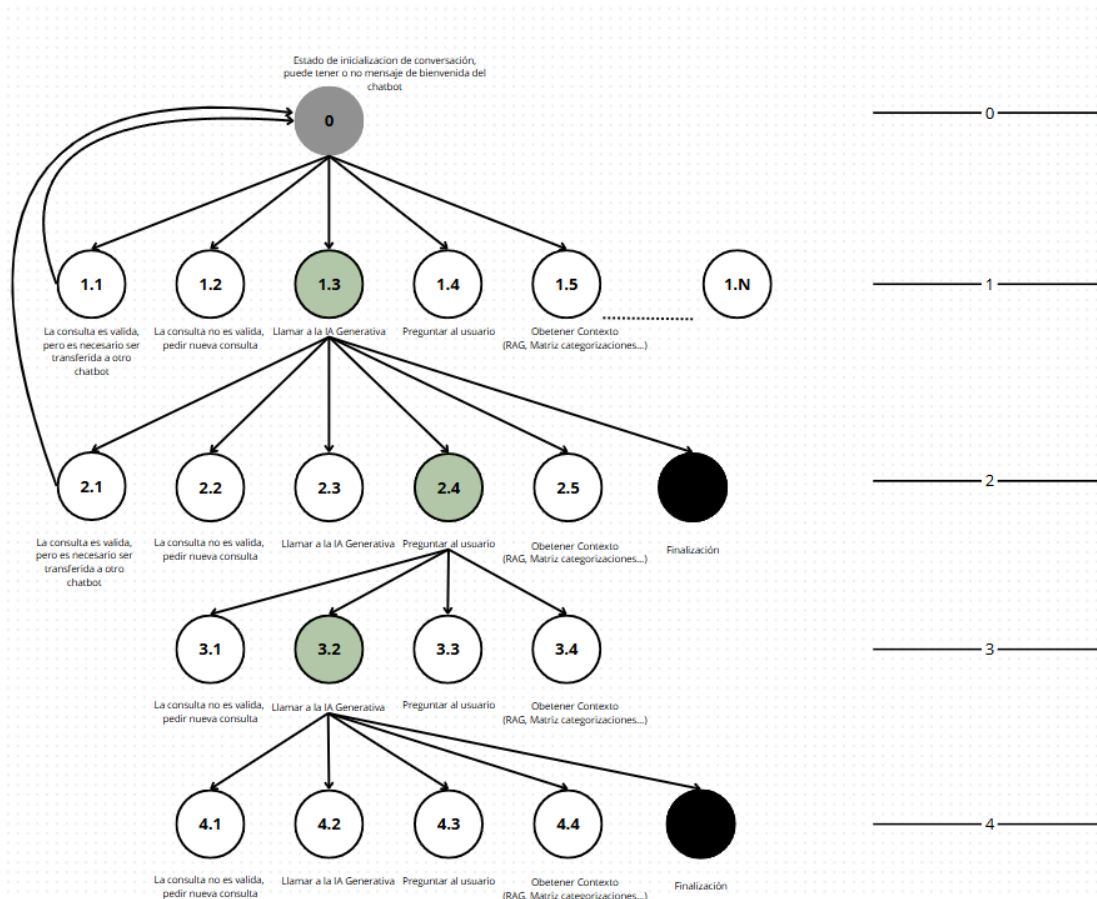
Inicialment, l'anàlisi es va centrar en una representació **altament abstracta**, basada en la dinàmica observable des del punt de vista de l'usuari. Aquesta aproximació de tipus **caixa negra** considerava la conversa com una seqüència de **interaccions de consulta-resposta**, on cada missatge de l'usuari rebia una resposta per part del bot. Aquesta visió simplificada va permetre caracteritzar les converses com a fils de comunicació entre dues parts, i posar en relleu la naturalesa iterativa del procés: l'usuari consulta, el bot respon, i es repeteix.



A mesura que es va avançar en l'anàlisi del comportament del chatbot, es va passar de la visió externa de caixa negra a un enfocament més detallat i intern, fent **zoom sobre les unitats que componen el seu flux conversacional**. Aquest procés va permetre **definir amb precisió què és un "pas" o "estat" dins del sistema**, entenent-lo com un punt concret del flux on el bot executa una acció específica: mostrar un missatge, esperar una resposta, fer una consulta, o transferir la conversa. Un pas no representa només un bloc de text, sinó una configuració completa que determina **què diu el bot, com ho diu i què espera que passi després**. Així, el chatbot pot **presentar un text al usuari en forma de pregunta si necessita aclariments** (pregunta oberta, binària, multiselecció...), **en forma de resposta directa generada per un model d'IA generativa si considera que té prou context, o pot fins i tot indicar que necessita una nova consulta si la prèvia no és tractable**. També pot actuar de manera autònoma consultant fonts externes (com un sistema RAG o una matriu de categoritzacions) per complementar la seva resposta abans d'interactuar amb l'usuari. Aquesta varietat d'accions dins d'un pas concreta com es genera cada iteració de la conversa i quines opcions té disponibles el sistema per continuar o tancar la interacció.

A partir d'aquesta observació es va elaborar un **diagrama de fils (thread view)**, que representa una línia temporal en la qual es van alternant els missatges de l'usuari i del bot. Tanmateix, aquest model no reflectia encara les decisions internes ni els criteris que segueix el bot per escollir la seva resposta, fet que va motivar la necessitat d'un nivell d'anàlisi més profund.

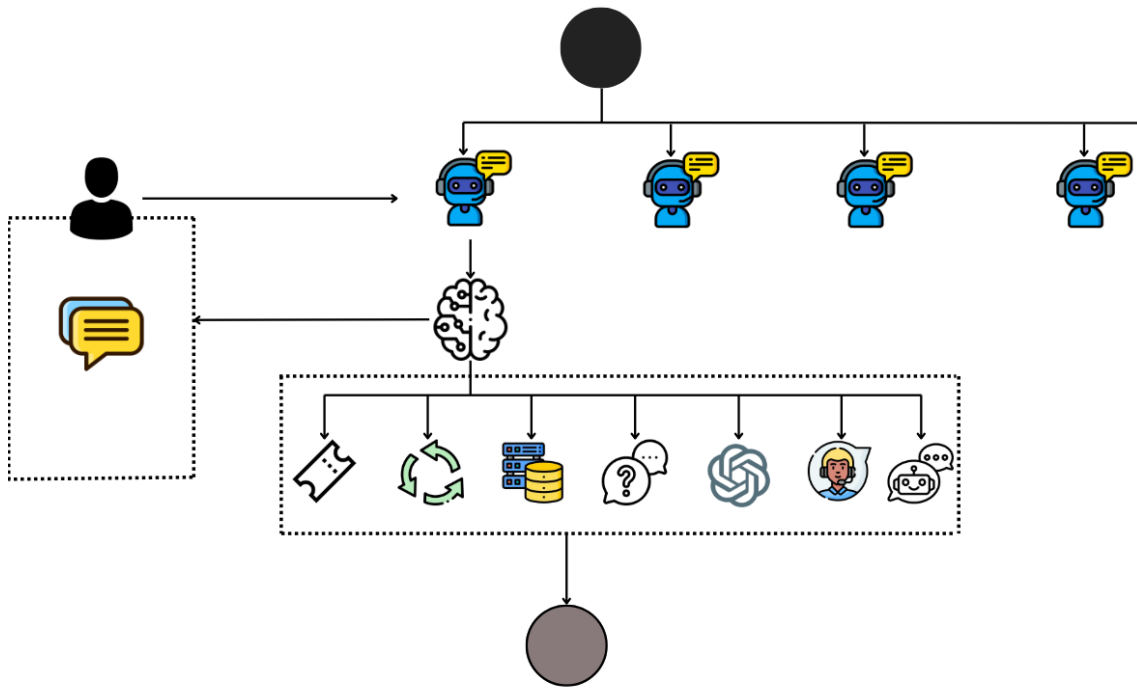
Així, es va procedir a estudiar el **flux intern del bot com si fos un fil d'execució** que recorre diferents fases o estats: rebre la consulta, analitzar-la, decidir com processar-la (amb IA, amb consulta a una base de dades, amb preguntes aclaridores, etc.), i finalment generar una resposta.



No obstant això, aquest model de flux lineal va resultar **insuficient**, ja que en la pràctica el bot pot **reentrar a fases anteriors** en funció de la qualitat o claredat de la resposta de l'usuari. Aquest comportament genera un **bucle entre fases**, especialment entre la fase de preguntar més informació i la d'analitzar de nou la consulta.

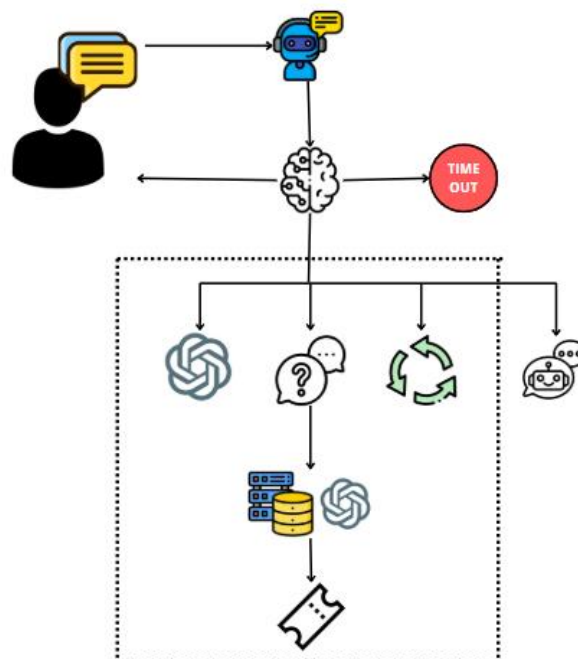
Per aquest motiu es va optar per una tercera representació: un **model basat en interaccions completes amb el focus posat en l'usuari**, on cada iteració completa del bot es considera **tancada en si mateixa**, atès que no es pot predir si l'usuari contestarà o donarà per finalitzada la conversa. D'aquesta manera, **cada resposta del bot es considera un punt de transició**, on la conversa pot acabar o continuar, depenent de si hi ha nova entrada per part de l'usuari.

Dins d'aquest esquema, es va identificar la necessitat de modelar explícitament un component funcional intern, anomenat informalment "**cervell**" del chatbot, que és el responsable de **decidir com s'ha de gestionar cada entrada rebuda per part de l'usuari**. Aquest mòdul de decisió no forma part directa del flux de passos, però és essencial per orquestrar-lo: **analitza el text rebut i determina quina estratègia de resposta és més adequada**. Aquesta estratègia pot consistir en generar una resposta mitjançant IA generativa, formular una pregunta per obtenir més informació, accedir a una base de dades, recuperar informació de context o fins i tot transferir la conversa a un altre sistema. El cervell no només selecciona el pas següent, sinó també la **eina o metodologia amb la qual s'ha d'emetre la resposta**, en funció del tipus de consulta, la claredat del missatge o el coneixement disponible. Aquest component és el que converteix un flux rígid en una arquitectura dinàmica, on cada iteració pot adaptar-se intel·ligentment a l'usuari, als recursos disponibles i a la complexitat de la sol·licitud.



A més, aquest enfocament va permetre incorporar la possibilitat que **el chatbot actual decideixi transferir la conversa a un altre chatbot especialitzat**, si detecta que la consulta no entra dins del seu domini competencial. Això introdueix una bifurcació important en el flux: el bot pot no només decidir quin pas intern seguir, sinó també si ha d'abandonar el control de la conversa i delegar-la a un altre sistema, iniciant una nova sessió dins un altre context.

Un cop plantejat aquest diagrama d'interaccions entre el chatbot i un usuari, es va tornar a fer el diagrama de com seria el flux d'un chatbot en concret, on te el "cervell" per decidir l'acció o pas al que vol transitar i la jerarquia d'aquests passos que pot fer per no repetir-se, per exempl: un cop ja s'ha intentat respondre amb IA Generativa i no ha funcionat, passar a la Base dades.



4.2.3. Taules d'anàlisi i transicions

Amb l'objectiu de formalitzar i validar el disseny del flux conversacional del chatbot, es va optar per crear un conjunt estructurat de taules en format Excel que permetessin descriure de manera clara i editable cadascun dels passos que el sistema pot executar. Aquestes taules constitueixen una representació semi-formal del funcionament intern del bot i serveixen tant com a document de disseny com a base per a la posterior implementació en base de dades o codi. La primera d'aquestes taules recull el catàleg complet de passos definits, indicant per a cadascun d'ells el missatge associat, el tipus de pas (pregunta, acció, resposta), si és un pas inicial o final, i el mètode amb què el bot genera la seva resposta (IA generativa, missatge predefinit, consulta a base de dades, etc.).

A més, es va construir una taula de transicions que especifica, per a cada pas, quins són els seus possibles passos següents, en funció de la condició que es compleixi. Aquestes condicions poden provenir de regles predefinides, classificacions fetes per un model d'IA, o interaccions explícites amb l'usuari. Per complementar aquest model, també s'inclou una taula de protocols que identifica les fonts d'informació o eines que pot emprar el bot en cada pas (com RAG, matriu de categoritzacions, o derivació a un altre sistema). Aquest conjunt de taules permet analitzar el sistema com una màquina d'estats parametrizable, i serveix com a suport per validar els fluxos, detectar inconsistències i fer simulacions de comportament abans de la seva execució real.

Taula de Passos

PASOS						
PASO	MENSAJE	TIPO DE MENSAJE	TIPO DE PASO	REQUIERE RESPUESTA	PASO INICIAL	PASO FINAL
Inicializar Conversación	Hola, ¿en qué puedo ayudarte respecto a X? // ""	Predefinito	Inicialización	True	True	False
Decisión acción	False	NO	Decisión	False	True	False
Pedir nueva consulta	Lo siento, no puedo responder a eso. ¿Puedes reformular tu pregunta?	Predefinito	Interacción con el usuario	True	False	True
Derivar a otro chatbot	Voy a transferirte al chatbot correspondiente...	Predefinito	Acción final	False	False	True
Consultar RAG	False	NO	Obtener Contexto	False	False	False
Responder con una IAGen	Respuesta de la IA Generativa	IA Generativa	Interacción con el usuario	True	False	True
Matriu Categoritzacions	False	NO	Obtenir Contexte	False	False	False
Pregunta de SI/No	¿Es correcto que te refieres a un problema de acceso? (SI/No)	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Abierta	¿Puedes darme más detalles sobre tu problema?	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Multiselección	Selecciona una de las opciones: 1, 2, 3...	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Intervalo	En qué rango de valores se encuentra tu problema?	IA Generativa	Interacción con el usuario	True	False	True
Paso Finalización	Gracias por usar el chatbot, ¡hasta luego!	Predefinito	Acción final	False	False	True
Contactar con agente humano	Te conectaré con un agente humano para continuar con tu solicitud.	Predefinito	Acción final	True	False	True
Generar Ticket de JIRA	Se ha generado un ticket en JIRA para hacer seguimiento a tu incidencia.	Predefinito	Acción final	True	False	True
Consultar base de datos interna	False	NO	Obtener Contexto	False	False	False

Aquesta taula descriu tots els possibles passos o estats que pot recórrer el chatbot durant la interacció amb l'usuari. Serveix com a referència per entendre la funció de cada pas, el seu comportament i el tipus de resposta que genera.

Columnes:

- **PAS:** Nom identificador del pas dins del flux conversacional.
- **MISSATGE:** Missatge que el bot mostra a l'usuari quan s'executa aquest pas. Pot ser un missatge predefinit, generat per IA, o en blanc si no s'envia res.
- **TIPUS DE MISSATGE:** Indica si el missatge és predefinit, generat per IA, una pregunta o si no s'utilitza cap missatge.
- **TIPUS DE PAS:** Classificació funcional del pas (inicialització, acció final, interacció, consulta, etc.).
- **REQUIEREIX RESPUESTA:** Especifica si el bot espera una resposta per part de l'usuari abans de continuar.
- **MÈTODE DE RESPUESTA:** Mecanisme mitjançant el qual el bot construeix la seva resposta (missatge predefinit, consulta a IA, accés a una font externa...).
- **PAS INICIAL:** Indica si el pas pot iniciar una conversa.
- **PAS FINAL:** Indica si el pas representa el estat en el que el chatbot està a espera de tornar a rebre una consulta de l'usuari per continuar amb la conversa, sino es considera com finalitzat.

A continuació, es presenten els diferents **passos** que pot seguir un chatbot durant una conversa amb l'usuari, classificats segons el **tipus de pas** que representen dins el flux de comportament. Aquesta classificació permet entendre la funció específica de cada pas dins l'arquitectura general del sistema, ja sigui com a punt d'inici, acció final, interacció directa amb l'usuari, o com a mecanisme intern per prendre decisions o recuperar informació. Aquesta distinció és clau per al disseny estructurat del flux conversacional i per assegurar un funcionament coherent i escalable del chatbot.

Inicialització

Aquest tipus de pas és responsable d'iniciar la conversa amb l'usuari. Es tracta d'un punt d'entrada al sistema, en el qual es genera un missatge de benvinguda i s'espera una primera entrada per part de l'usuari.

- *Inicialitzar Conversació – envia una salutació o activa la captura de la primera consulta.*

Decisió

Els passos de tipus decisió tenen la funció de processar la consulta de l'usuari i determinar, mitjançant regles, classificadors o models d'IA, quin serà el següent pas. No mostren cap missatge a l'usuari i no requereixen resposta. Actuen com a "cervell" del sistema.

- *Decidir acció – analitza la intenció de l'usuari i decideix si s'ha de preguntar més, consultar dades, o respondre directament.*

Interacció amb l'usuari

Aquests passos es caracteritzen per generar un missatge dirigit a l'usuari i **esperar una resposta** per continuar el flux. Sovint s'utilitzen per afinar la consulta, recollir informació addicional o confirmar dades. Els missatges poden ser generats per IA o seguir plantilles predefinides.

- *Pregunta de Sí/No – confirmació binària.*
- *Pregunta Multiselecció – tria d'opcions.*
- *Pregunta Interval – valor quantitatiu.*
- *Pregunta Oberta – resposta lliure.*
- *Respondre amb una IAGen – ofereix resposta generada per IA i espera si cal seguir.*
- *Obtenir Context - Aquest tipus de pas s'utilitza quan el sistema ha de recuperar informació externa o de suport abans de poder continuar amb la conversa. No genera cap resposta per l'usuari de manera directa, sinó que actua com a connector amb sistemes de coneixement (com RAG o Matriu de Categoritzacions).*

Acció final

Són passos que **no requereixen cap resposta** i que tanquen la conversa o executen una acció concreta i final. Això inclou transferències a humans, generació de tickets o simplement finalització formal del diàleg.

- *Derivar a un altre chatbot*
- *Contactar amb agent humà*
- *Generar Ticket de JIRA*
- *Pas Finalització*

Taula de Transicions

TRANSICIONES		
PASO	PASO SIGUIENTE	TIPO DE TRANSICIÓN
Inicializar Conversación	Decidir acción	Interactuar con el usuario
Decidir acción	Pedir nueva consulta	Regla predefinida
Decidir acción	Derivar a otro chatbot	Regla predefinida
Decidir acción	Consultar RAG	Sistema
Decidir acción	Responder con una IAGen	Sistema
Decidir acción	Matriz Categorizaciones	Sistema
Decidir acción	Pregunta de Si/No	Sistema
Decidir acción	Pregunta Abierta	Sistema
Decidir acción	Pregunta Multiselección	Sistema
Decidir acción	Pregunta Intervalo	Sistema
Pedir nueva consulta	Decidir acción	Sistema
Derivar a otro chatbot	Inicializar Conversación	Sistema
Responder con una IAGen	Paso Finalización	Sistema\Interacción
Responder con una IAGen	Decidir acción	Interactuar con el usuario
Pregunta de Si/No	Decidir acción	Interactuar con el usuario
Pregunta Abierta	Decidir acción	Interactuar con el usuario
Pregunta Multiselección	Decidir acción	Interactuar con el usuario
Pregunta Intervalo	Decidir acción	Interactuar con el usuario
Matriz Categorizaciones	Responder con una IAGen	Sistema
Decidir acción	Contactar con agente humano	Interactuar con el usuario
Decidir acción	Generar Ticket de JIRA	Interactuar con el usuario
Decidir acción	Consultar base de datos interna	Sistema
Consultar RAG	Responder con una IAGen	Sistema
Consultar base de datos interna	Responder con una IAGen	Sistema

La taula de transicions modela com es passa d'un pas a un altre dins el flux del chatbot. Cada fila representa una transició possible, amb la seva lògica associada.

Columnes:

- PAS: Pas d'origen des d'on parteix la transició.
- PAS SEGÜENT: Pas de destinació al qual es transita si es compleix la condició.
- TIPUS DE TRANSICIÓ: Tipologia de la transició (sistema (ho fa el chatbot de forma automàtica), per regla, interacció...).

Taula de Condicions

CONDICIONES			
PASO INICIAL	CONDICIÓN	FUENTE DE LA CONDICION	PASO SIGUIENTE
Inicializar Conversación	Se recibe la consulta del usuario.	Interactuar con el usuario	Decidir acción
Decidir acción	Si la consulta no puede ser respondida y el usuario debe reformularla.	Regla predefinida	Pedir nueva consulta
Decidir acción	Si la consulta pertenece a otro chatbot especializado.	Regla predefinida	Derivar a otro chatbot
Decidir acción	Si el sistema determina que no tiene suficiente información y se necesita contexto externo.	Sistema	Consultar RAG
Decidir acción	Si el sistema determina que la consulta es clara y puede ser respondida por la IA Generativa.	Sistema	Responder
Decidir acción	Si el sistema determina que la respuesta requiere definir un rango de valores.	Sistema	Pregunta Intervalo
Decidir acción	Si el sistema determina que la IA necesita una confirmación binaria para continuar.	Sistema	Pregunta de Si/No
Decidir acción	Si el sistema determina que hay opciones predefinidas y el usuario debe elegir una.	Sistema	Pregunta Multiselección
Decidir acción	Si el sistema determina que la consulta es abierta y necesita más detalles del usuario.	Sistema	Pregunta Abierta
Pedir nueva consulta	Si el usuario ha reformulado la consulta.	Interactuar con el usuario	Decidir acción
Derivar a otro chatbot	Si el chatbot de destino está disponible.	False	Finalizar ciclo
Responder con una IAGen	Si el usuario no hace mas consultas o ya esta satisfecho	Interactuar con el usuario	Paso Finalización
Responder con una IAGen	Si el usuario hace otra consulta	Interactuar con el usuario	Decidir acción
Pregunta de Si/No	Si el usuario responde con 'Si' o 'No', se evalúa la transición.	Interactuar con el usuario	Decidir acción
Pregunta Abierta	Si el usuario proporciona una respuesta que permite clasificar la consulta.	Interactuar con el usuario	Decidir acción
Pregunta Multiselección	Si el usuario selecciona una opción válida.	Interactuar con el usuario	Decidir acción
Pregunta Intervalo	Si el usuario proporciona un valor dentro del intervalo esperado.	Interactuar con el usuario	Decidir acción
Matriz Categorizaciones	Si ya se han recuperado los datos necesarios para dar una respuesta con la IA Generativa	Sistema	Responder con una IAGen
Consultar base de datos interna	Si se recuperan los datos necesarios de la base interna, se pasa a la IA Generativa	Sistema	Responder con una IAGen
Consultar RAG	Si se recuperan los fragmentos necesarios para responder con la IA Generativa	Sistema	Responder con una IAGen
Decidir acción	Si el sistema decide que necesita recuperar el protocolo de la Matriz de Categorizaciones	Sistema	Matriz Categorizaciones
Decidir acción	Si el sistema determina que la consulta requiere asistencia humana	Sistema	Contactar con agente humano
Decidir acción	Si el sistema determina que la consulta requiere generar una incidencia en JIRA	Sistema	Generar Ticket de JIRA
Decidir acción	Si el sistema determina que la información debe obtenerse de la base de datos interna	Sistema	Consultar base de datos interna

Aquesta taula detalla les condicions específiques que determinen cap a quin pas es dirigeix el sistema des d'un pas actual. És clau per entendre el comportament intel·ligent i adaptatiu del bot.

Columnes:

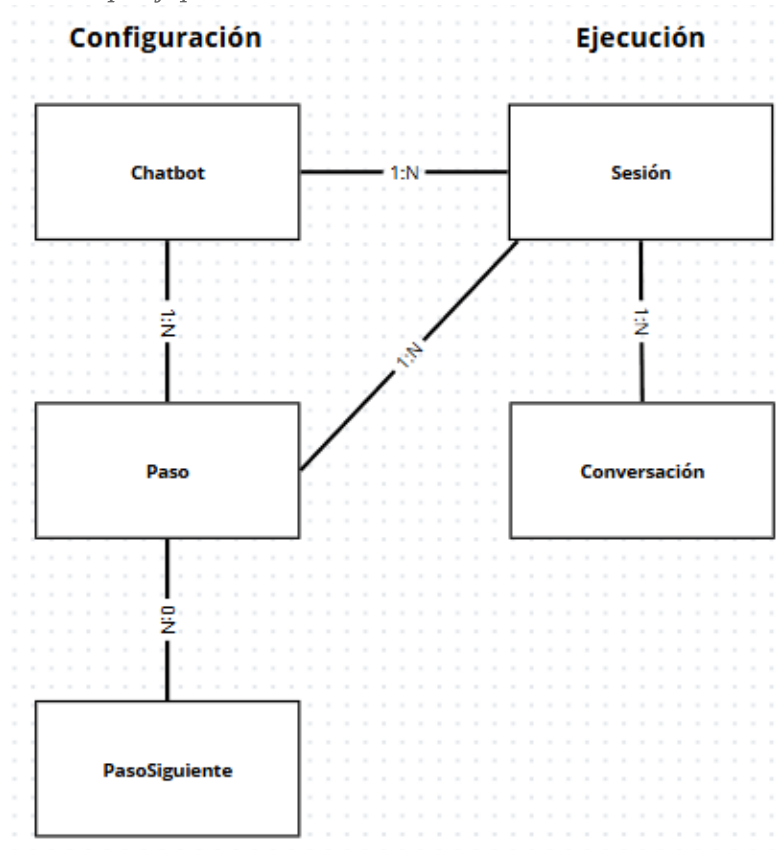
- **PAS INICIAL:** Pas actual on s'avaluarà la condició.
- **CONDICIÓ:** Criteri o situació que ha de complir-se per activar la transició.
- **FONT DE LA CONDICIÓN:** Origen de la condició (interacció amb l'usuari, model d'IA, regla predefinida...).
- **PAS SEGÜENT:** Pas al qual es transita si la condició s'acompleix.

4.2.4. Modelatge de la base de dades

Amb l'objectiu de sustentar tècnicament el funcionament del chatbot dins un entorn modular i escalable, s'ha dissenyat una base de dades que permet gestionar tant la **configuració estructural del sistema** com la **traçabilitat de les interaccions** durant l'execució. Aquest disseny segueix una metodologia estructurada amb tres fases de representació: el **model entitat-relació (E-R)**, el **model lògic relacional**, i el **model físic**, el qual es troba actualment en procés de desenvolupament.

4.2.4.1. Disseny del Model Entitat-Relació

A continuació es presenta una explicació detallada del model entitat-relació (E-R) desenvolupat per a representar l'estructura de dades d'un sistema de chatbot modular. Aquest model té com a objectiu capturar, de manera estructurada i escalable, tant la lògica de configuració del comportament del chatbot com les dades associades a la seva execució en temps real. El model es divideix conceptualment en dues grans àrees: **Configuració i Execució**, cadascuna amb les seves entitats i relacions específiques.



4.2.4.1.1. Zona de Configuració

Aquesta part del model representa la definició estructural i estàtica del comportament del chatbot, és a dir, la manera com es configura el flux conversacional, independentment de les converses específiques que es duguin a terme. Inclou les següents entitats:

Chatbot

És l'entitat central que representa un bot concret dins del sistema. Cada chatbot pot tenir una configuració pròpia de passos, regles i protocols, de manera que es pugui adaptar a diferents casos d'ús o àrees funcionals (incidències, recursos humans, consultes tècniques, etc.).

- Un Chatbot pot tenir múltiples Pasos definits (relació 1:N).
- Un Chatbot també pot estar associat a diverses Sessions d'execució (relació 1:N), ja que pot atendre diversos usuaris en moments diferents.

Paso

Representa un estat o acció dins del flux conversacional. Cada pas pot ser un missatge, una consulta, una acció de derivació, una pregunta al sistema o una petició a una font de dades externa. A més, cada pas té atributs associats com el tipus de pas (interacció, consulta de context, acció final...), si és pas inicial o final, si requereix resposta, etc.

- Cada Paso està vinculat a un Chatbot (relació N:1).
- Cada Paso pot tenir diverses sortides possibles a altres passos a través de l'entitat PasoSiguiende (relació 0:N).

PasoSiguiende

És una entitat auxiliar que modela la transició condicional entre passos. Permet representar que un pas pot conduir a múltiples passos següents, depenent de condicions específiques (resposta de l'usuari, resultat d'una consulta, regla definida...).

- Aquesta entitat connecta un Paso amb un altre Paso, i pot contenir informació extra com la condició de transició.
- Relació entre Paso i PasoSiguiende: 1:N (un pas pot tenir molts passos següents).

4.2.4.1.2. Zona d'Execució

Aquesta part del model està enfocada a registrar l'activitat real del sistema en temps d'execució, és a dir, tot allò que passa quan un usuari interactua amb un chatbot. Les entitats d'aquesta secció permeten fer seguiment de les converses, les sessions i els missatges intercanviats.

Sesión

Una sessió representa un període d'interacció activa entre un usuari i un chatbot concret. Té una temporalitat definida i pot contenir múltiples converses. Aquesta entitat serveix per agrupar i organitzar les converses relacionades entre si, especialment útil en casos de transferències entre bots o d'interaccions recurrents.

- Un Chatbot pot tenir múltiples Sessions associades (relació 1:N).
- Una Sessió pot contenir múltiples Conversaciones (relació 1:N).

Conversación

Representa un bloc de comunicació coherent dins d'una sessió. Per exemple, una única petició de l'usuari i les respostes que es generen com a resultat. Es poden registrar com a converses separades dins d'una mateixa sessió si hi ha canvis d'intenció o intervals de temps prolongats.

- Cada Sessió pot tenir diverses Conversaciones associades (relació 1:N).

4.2.4.2. Disseny del model lògic de dades

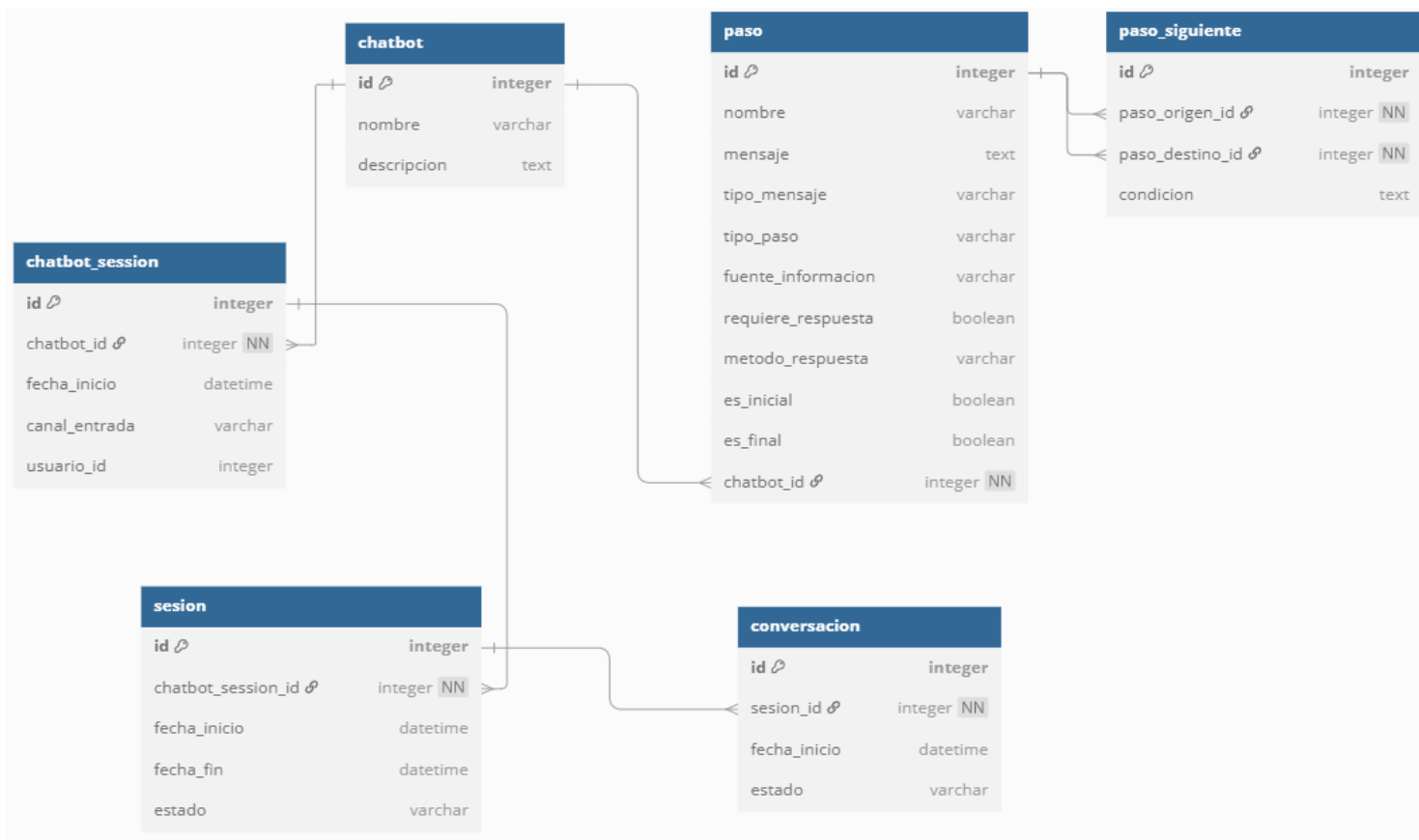
En aquesta etapa del projecte, s'ha dut a terme la definició formal del model lògic de dades que sustenta el funcionament del sistema de gestió de converses del chatbot. Aquest model representa una evolució directa del model conceptual (E-R) i reflecteix de manera estructurada com es materialitzen les entitats, les seves relacions i els atributs clau dins una arquitectura relacional.

El diagrama lògic s'ha dissenyat amb l'objectiu de cobrir tant els aspectes de configuració del comportament del chatbot com l'execució de les converses reals amb els usuaris. D'aquesta manera, el model queda organitzat en dues branques principals:

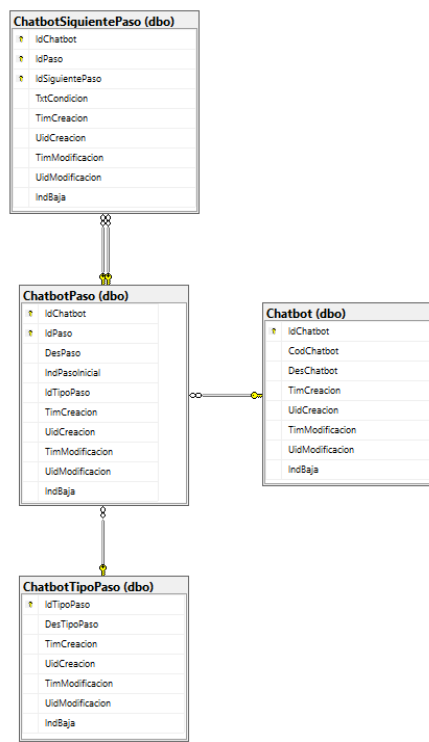
- **Bústia de configuració:** inclou les entitats chatbot, paso i paso_siguiete. Aquestes taules representen l'estructura fixa de cada chatbot: les preguntes, possibles respostes, tipus d'interaccions i les condicions que governen les transicions entre passos. Són, per tant, taules de referència que modelen el comportament programat del sistema.
- **Bústia d'execució:** composta per chatbot_session, sesion i conversacion. Aquestes entitats recullen l'activitat dinàmica generada durant la interacció real amb els usuaris. Així, cada vegada que un usuari accedeix al servei, es crea una nova instància de chatbot_session, la qual pot contenir una o diverses sesion, que al seu torn poden incloure múltiples conversacion separades segons el context o el moment de la interacció.

Aquest model també reflecteix l'estructura jeràrquica natural de les dades:

- Un chatbot pot tenir molts passos, i cada pas pot tenir diverses transicions definides a paso_siguiete.
- Cada chatbot_session està vinculada a un chatbot concret i representa l'entrada d'un usuari en un canal determinat.
- Una sesion forma part d'una chatbot_session i defineix un tram d'interacció viva amb l'usuari.
- Cada conversacion s'assigna a una sessió i recull el contingut o context específic d'una consulta.



Cal destacar que el model físic derivat d'aquest diagrama es troba actualment en desenvolupament, i incorporarà optimitzacions addicionals com clau foranes, índexs i normalització, així com possibles taules complementàries per al registre d'activitat o auditories, pero actualment es pot mostrar el estat actual d'aquest model físic



4.3. PoC: Chatbot de gestió d'incidències

En aquesta secció es presenta l'arquitectura i el comportament del Proof of Concept (PoC) del chatbot de gestió d'incidències. El sistema s'ha dissenyat seguint un enfocament seqüencial sense un cervell central, on el flux avança a través d'una sèrie de passos predefinits i protocols específics. Cada pas es troba estrictament condicionat a la informació recollida prèviament i a la resposta de l'usuari, garantint així una traçabilitat completa de la conversa i una gestió eficient de cada cas.

Cal destacar que la part del desenvolupament del **front-end** està sent gestionada pel **departament d'Aplicacions Col·laboratives**, que s'encarrega de construir la interfície d'usuari i d'integrar-la amb el motor del chatbot desenvolupat. Aquesta col·laboració permet mantenir una separació clara entre el desenvolupament de la lògica del flux conversacional i la seva presentació visual al usuari final.

4.3.1. Funcionament del PoC i els seus prompts

Aquest comportament seqüencial es basa en una sèrie de prompts preconfigurats que s'activen segons el context i l'estat de la conversa. Cada prompt té un objectiu concret dins del flux: detectar si l'usuari està plantejant una incidència, demanar informació complementària, suggerir accions de resolució, classificar la incidència segons una llista predefinida, o bé derivar-la a sistemes externs com Jira en cas necessari. El sistema alterna entre aquests prompts en funció de l'estat del problema, del coneixement previ, i del comportament de l'usuari durant la conversa.

A continuació, es presenta una taula que recull els prompts principals utilitzats en el sistema i la seva funció dins del flux d'actuació:

Prompt	Funció dins del flux del PoC
<code>detectar_incidencia_prompt</code>	Detecta si el missatge de l'usuari descriu una incidència real o és massa genèric.
<code>microsoft_prompt</code>	Resol les incidències senzilles dins del domini Microsoft i valida si la incidència s'ha solucionat.
<code>info_prompt</code>	Demana informació específica necessària per continuar amb la resolució.
<code>action_prompt</code>	Sol·licita a l'usuari que realitzi una acció per intentar resoldre la incidència.
<code>image_prompt</code>	Interpreta el contingut d'imatges enviades per l'usuari relacionades amb la incidència.
<code>classify_prompt</code>	Classifica la incidència segons una llista de descripcions predefinides per a escollir el protocol.
<code>jira_prompt</code>	Genera un ticket en el sistema JIRA amb la informació recopilada durant la conversa.
<code>jira_done</code>	Verifica si el resum del ticket generat en Jira és correcte o necessita ser modificat.

Actualment, amb el desenvolupament hem implementat perquè el PoC funcioni amb el `detectar_incidencia_prompt`, el `microsoft_prompt`, el `image_prompt`, aquesta es la seva descripció:

detectar_incidencia_prompt

Aquest prompt és responsable d'identificar si un missatge enviat per l'usuari conté la descripció d'una incidència informàtica. El comportament del sistema davant d'aquest prompt és binari: si detecta una incidència clara, retorna el marcador especial `<DONE>`, indicant que la consulta pot ser tractada. En cas contrari, genera una resposta educada informant l'usuari que es tracta d'un assistent virtual de resolució d'incidències i sol·licita una descripció més detallada del problema. Aquest prompt és essencial com a punt de partida del flux, ja que filtra interaccions no rellevants i assegura que només es processen aquelles que poden ser considerades veritables incidències.

microsoft_prompt

Aquest prompt és un dels components centrals del PoC, dissenyat específicament per gestionar incidències senzilles relacionades amb sistemes informàtics, excepte aquelles que involucren productes propis de l'empresa asseguradora o sistemes especialitzats. La seva resposta s'adapta segons l'estat de la conversa: pot indicar que no pot resoldre la incidència (`<NOT_DONE>`), pot proposar solucions concretes mitjançant una estructura específica ("Prueba las siguientes acciones...") i sol·licitar retroacció de l'usuari, o pot confirmar que la incidència ha estat resolta (`<DONE>`). Aquest prompt és fonamental per validar el comportament condicional del sistema i per integrar regles empresarials específiques dins del flux conversacional.

image_prompt

Aquest prompt s'activa quan l'usuari envia una imatge relacionada amb la seva incidència. El seu objectiu és generar una descripció clara, detallada i precisa del contingut de cada imatge, sempre en format de paràgraf únic per imatge. Aquest component és especialment útil per casos en què la incidència no pot ser expressada textualment o quan l'usuari complementa la seva explicació amb captures de pantalla o fotografies. D'aquesta manera, el sistema pot continuar el flux de resolució considerant elements visuals, afegint una capa d'anàlisi contextual visual al procés.

4.3.2. Funcionament del flux conversacional del PoC

El flux del Proof of Concept (PoC) desenvolupat per al chatbot de gestió d'incidències segueix una estructura seqüencial basada en passos predefinits i l'activació de diferents prompts segons l'estat de la consulta i el context disponible. Aquest flux no disposa d'un cervell central que orquestri tot el comportament del sistema, sinó que cada pas conté una lògica interna que decideix la transició següent en funció del resultat de la interacció actual.

Inici del flux

El procés s'inicia quan un usuari envia una consulta, que pot ser un simple missatge de text o incorporar elements més complexos com imatges o documents. El sistema primer comprova si ja existeix una conversa activa vinculada a aquell usuari. En cas afirmatiu, es recupera l'historial de la sessió per garantir la continuïtat i el context. Si no n'hi ha cap, es genera una nova sessió amb un identificador únic, on es registraran totes les interaccions futures.

Processament de la consulta

Quan el missatge entra al sistema, s'analitzen tots els components disponibles. Si la consulta inclou imatges, el sistema activa el `image_prompt`, que genera descripcions detallades per a cada

imatge i les integra dins el context de la consulta. Aquest context enriquit permet al model d'IA entendre millor la problemàtica plantejada.

A continuació, la conversa adquireix un **estat inicial**, que pot determinar si cal analitzar text, gestionar arxius adjunts o passar directament a la classificació. En aquest punt entra en joc el `detectar_incidència_prompt`, que actua com a primer filtre: si no es detecta una incidència informàtica clara, el sistema respon amb un missatge predefinit i finalitza el flux. En canvi, si es confirma la presència d'una incidència, es continua amb el procés de resolució.

Gestió de la consulta vàlida

En aquesta fase, el sistema pot intentar resoldre el problema directament. Per fer-ho, es fa ús del `microsoft_prompt`, que permet proporcionar respostes directes a problemes senzills si la consulta es troba dins l'àmbit de cobertura. Si no és així, s'activa el procés de classificació avançada, utilitzant el `classify_prompt`.

Aquest prompt classifica la consulta dins d'una categoria concreta mitjançant una matriu de categoritzacions prèviament definida. Per dur-ho a terme, es fa una crida al mòdul d'Analítica de Negoci, que retorna el protocol adequat. El sistema filtra l'Excel de categoritzacions amb aquest protocol, genera un prompt dinàmic amb les opcions disponibles i l'envia al model perquè retorni una categoria.

Casos especials i decisions del sistema

Els resultats d'aquesta classificació poden tenir dos efectes:

- Si la consulta ha estat classificada amb èxit, el chatbot pot respondre amb informació detallada, demanar informació addicional (`info_prompt`), o bé indicar una acció concreta a realitzar (`action_prompt`).
- Si la classificació retorna un valor invàlid (categoria 0), el sistema activa altres mecanismes com la creació automàtica d'un ticket de JIRA (`jira_prompt`) o la derivació a un altre equip.

Durant aquest procés, el sistema interpreta respostes amb valors especials com `<DONE>` o `<NOT_DONE>` per decidir les transicions:

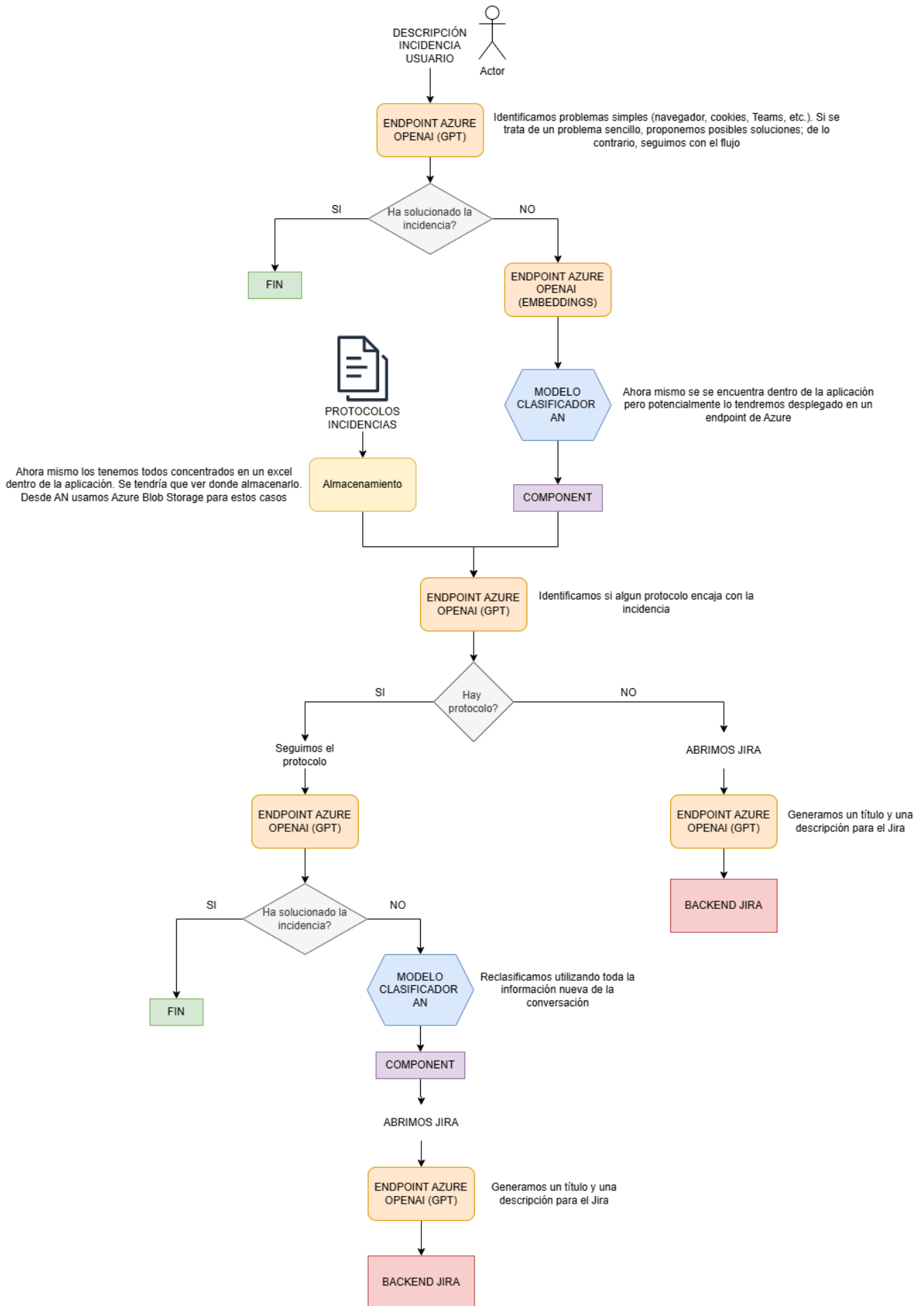
- `<DONE>` indica que la tasca ha estat completada (per exemple, s'ha resolt el problema o s'ha recollit tota la informació).
- `<NOT_DONE>` indica que cal continuar amb el flux o escalar el cas.

Si s'arriba a un punt on la incidència no es pot resoldre, es genera un **ticket de JIRA** amb tota la informació recopilada. Abans de crear el ticket, el sistema presenta al client un resum automatitzat del problema, i li demana que el validi (`jira_done`). Si el client vol modificar-lo, pot proporcionar una descripció alternativa.

Finalització del flux

Un cop la incidència ha estat resolta o escalada, el chatbot informa l'usuari de la situació. Si s'ha obert un ticket, s'inclou un enllaç per fer-ne el seguiment. Si s'ha resolt el problema directament, s'ofereix una resposta clara i es demana si necessita ajuda addicional.

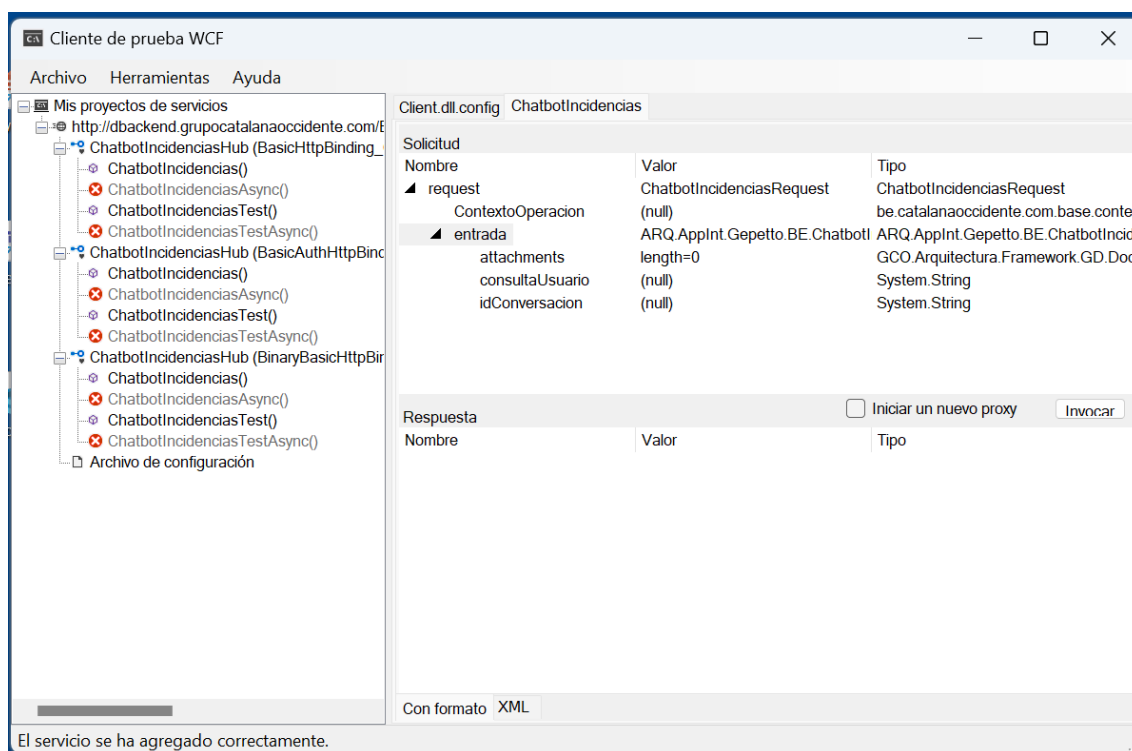
Finalment, tot i que el sistema no emmagatzema permanentment les converses, sí que utilitza una memòria temporal durant la sessió per mantenir el context actiu i garantir una experiència coherent.



4.3.3. Evidències del funcionament real

4.3.3.1. WCF Test Client

Per dur a terme la validació funcional del backend del chatbot, s'ha utilitzat l'eina **WCF Test Client** una aplicació gràfica proporcionada per Microsoft que permet consumir i provar serveis web (WCF Services) de forma controlada i repetible.



Aquesta eina ha estat essencial per validar l'execució de les funcionalitats implementades en el backend, especialment en les primeres fases del desenvolupament on encara no es disposava del front-end operatiu. El seu ús ha permès enviar peticions simulades, observar les respostes generades pel sistema i verificar que el comportament és l'esperat en diferents situacions.

Les captures de pantalla incloses en aquest apartat mostren exemples reals de cada prova, evidenciant el funcionament correcte de les primeres funcionalitats implementades al PoC.

4.3.3.2. Implementació tècnica dels prompts: detectar_incidencia_prompt i microsoft_prompt

Per tal de verificar que el comportament del chatbot desenvolupat en el Proof of Concept (PoC) és funcional i segueix el flux dissenyat, s'han obtingut evidències reals de l'execució del sistema mitjançant peticions simulades i respostes generades.

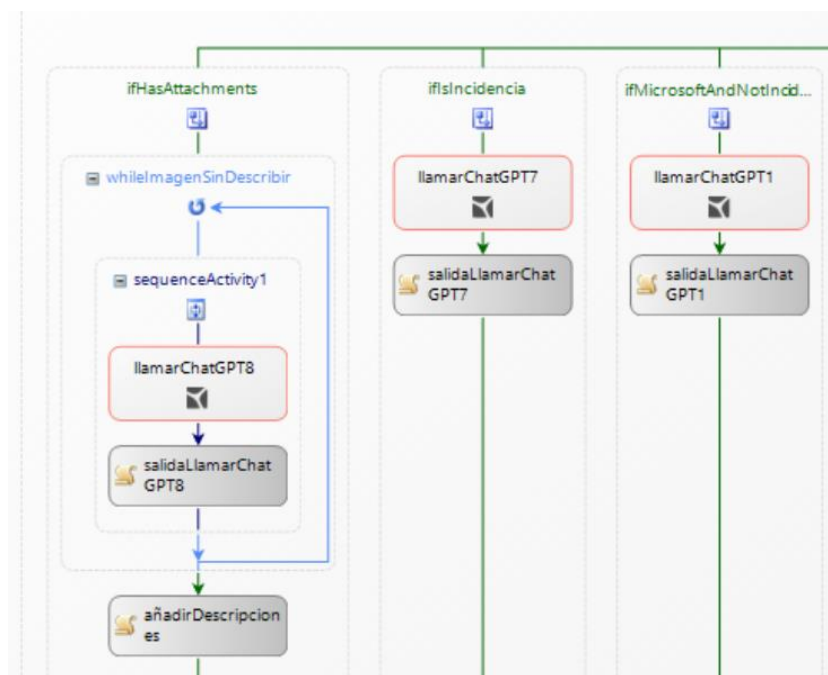
Aquestes evidències es centren en demostrar el correcte funcionament dels prompts ja desenvolupats, com ara `detectar_incidencia_prompt` i `microsoft_prompt`. A través de captures de pantalla s'il·lustra com el chatbot:

- Detecta si una consulta conté o no una descripció d'una incidència informàtica.
- Respon amb IA generativa utilitzant regles predefinides adaptades al domini Microsoft.

- Proporciona una resposta estructurada i adaptada al cas d'ús.
- Retorna senyals <DONE> o <NOT_DONE> en funció de la situació del problema i la interacció amb l'usuari.

Aquestes evidències són fonamentals per validar que el flux seqüencial del PoC és viable i que els prompts funcionen de forma efectiva i coherent dins del model d'interacció previst.

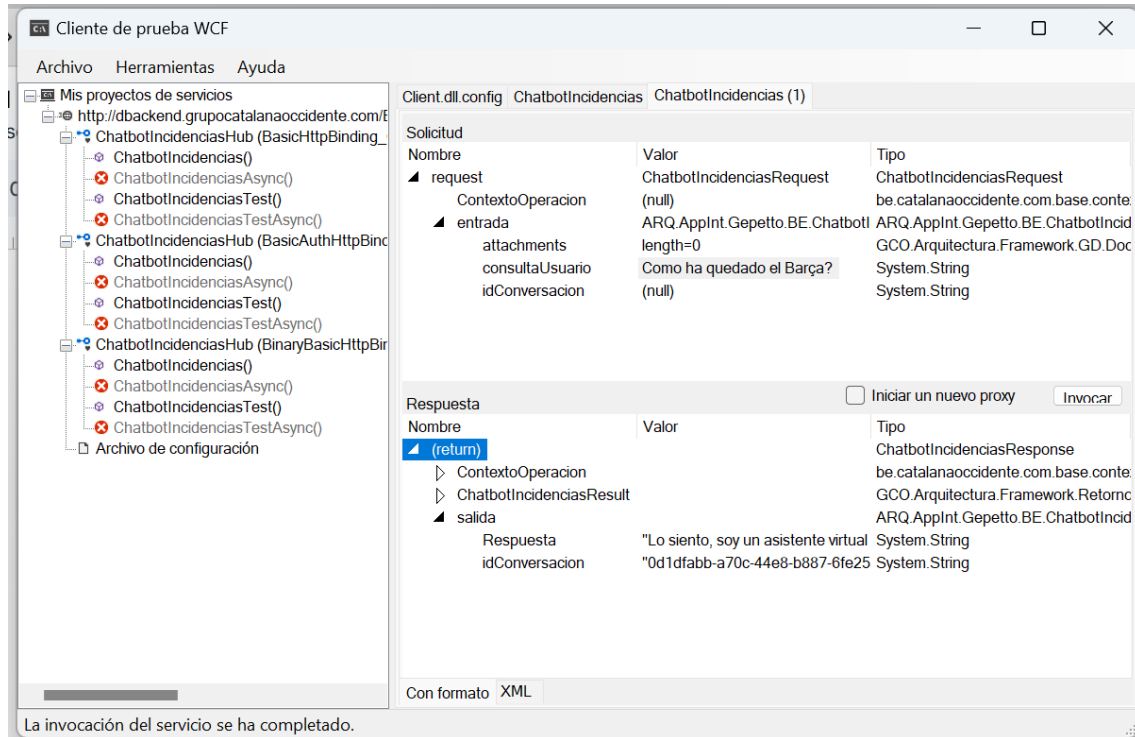
A nivell d'arquitectura, el desenvolupament del Proof of Concept del chatbot s'ha estructurat mitjançant una **activitat composta (Workflow Foundation)**, que representa una unitat de lògica de negoci encapsulada dins d'un flux de treball. Aquesta activitat, anomenada *ChatbotIncidencias*, és responsable de gestionar l'entrada del missatge, l'estat de la conversa, les crides als mòduls de processament (IA generativa, classificació, accions, etc.) i el retorn de la resposta adequada a l'usuari.



En aquesta activitat composta, s'utilitza una estructura de control anomenada *CacheStates* per mantenir els estats actius del flux de conversa, com ara si hi ha fitxers adjunts, si cal fer una classificació, si la incidència pot ser resolta amb un protocol conegut, o si cal escalar el cas a un sistema extern com JIRA.

Validació del detectar_incidencia_prompt

Per validar el funcionament del prompt detectar_incidencia_prompt, es va realitzar una primera prova amb la consulta: "**¿Cómo ha quedado el Barça?**", amb l'objectiu d'avaluar la resposta del sistema davant d'un missatge que no representa una incidència informàtica.



Com a resultat, es va observar que el sistema va gestionar correctament la petició:

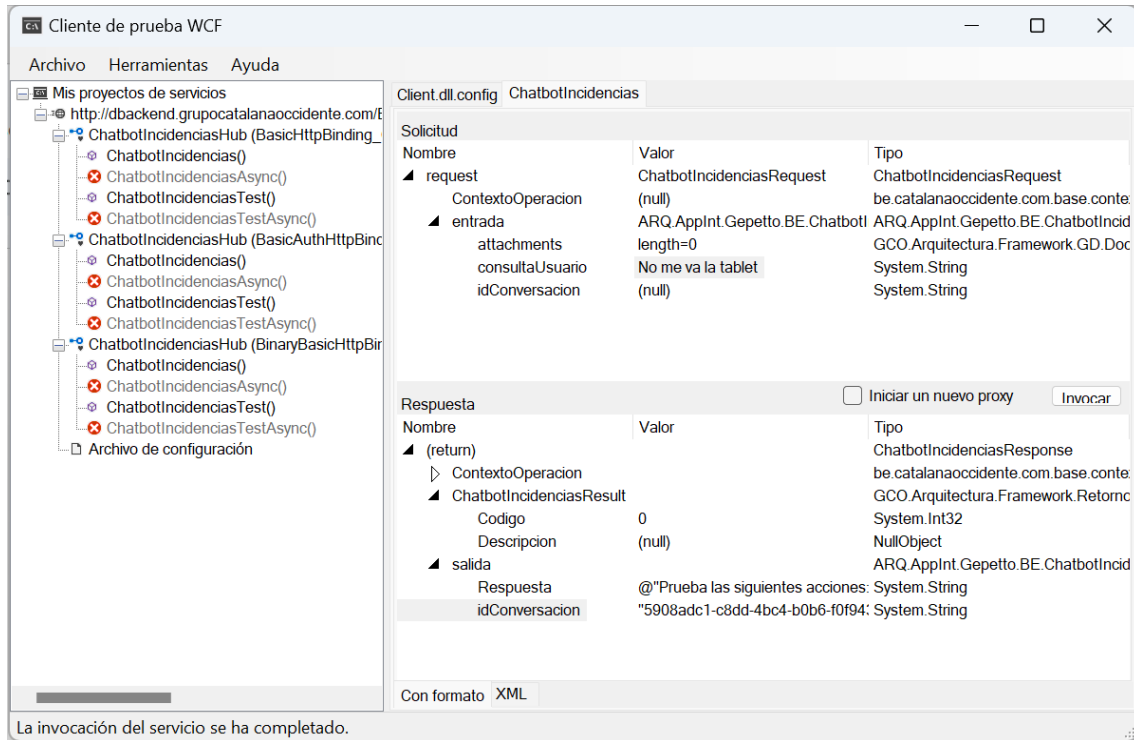
- Es va generar automàticament un **ID de conversa** per identificar el fil obert amb l'usuari.
- El chatbot va emetre la següent resposta predefinida, coherent amb el comportament esperat segons el prompt establert:

"Lo siento, soy un asistente virtual de resolución de incidencias informáticas. Si tienes alguna incidencia con tu equipo o software, por favor descríbela lo mejor posible para que pueda ayudarte."

Aquesta resposta mostra que el sistema ha identificat correctament que el missatge no era una incidència, i ha reconduït l'usuari cap al propòsit del chatbot de forma clara i educada, complint amb el que especifica el prompt de detecció.

Validació del microsoft_prompt

Per verificar el funcionament del prompt microsoft_prompt, es va simular una consulta tècnica habitual amb el missatge: **"No me va la tablet"**, amb l'objectiu de comprovar si el chatbot era capaç de proporcionar instruccions clares i útils per intentar resoldre el problema de forma autònoma.



El sistema, un cop confirmada la naturalesa d'incidència del missatge mitjançant el prompt anterior (detectar_incidencia_prompt), va procedir automàticament a activar el prompt de diagnosi inicial. La resposta generada per la IA va ser coherent i adaptada a un usuari sense coneixements tècnics, seguint fidelment el comportament esperat:

"@Prueba las siguientes acciones:

1. Asegúrate de que la tablet esté cargada. Conéctala a la corriente y espera unos minutos antes de intentar encenderla de nuevo.
 2. Si la tablet está cargada pero no se enciende, intenta reiniciarla. Para hacerlo, mantén pulsado el botón de encendido durante unos 10 segundos hasta que la tablet se apague y luego se vuelva a encender.
 3. Si la tablet se enciende pero no funciona correctamente, puede que necesites actualizar el sistema operativo. Para hacerlo, ve a "Ajustes", luego a "Sistema" y finalmente a "Actualización de software". Si hay una actualización disponible, instálala.
- ¿Te ha solucionado el problema?"

Aquesta resposta compleix amb tots els criteris funcionals definits en el prompt: ofereix passos clars i seqüencials per intentar solucionar el problema, i acaba amb una pregunta que permet al chatbot saber si s'ha resolt la incidència. El missatge ha estat adaptat correctament per a un públic no tècnic i no inclou cap derivació a altres canals, tal com estava especificat.

Limitació en la validació del processament d'arxius adjunts

*Tot i que el sistema ja contempla la gestió de fitxers adjunts (com imatges o documents), actualment no ha estat possible validar aquest comportament, ja que depèn del desplegament complet del **portal web d'usuari**. Aquest portal serà l'encarregat de permetre als usuaris adjuntar documents i enviar-los juntament amb les consultes. Un cop operatiu, es podrà testar l'activació del `image_prompt`, la generació de descripcions automàtiques per a cada arxiu i la seva correcta integració dins del context de la conversa. Aquesta validació quedarà pendent per a una fase posterior del projecte.*

4.3.4. Conclusions visuals i tècniques del bloc

El treball realitzat durant aquest bloc ha tingut un impacte notable tant a nivell conceptual com tècnic. A través de l'anàlisi inicial, s'ha pogut entendre amb profunditat el funcionament intern d'un chatbot intel·ligent i la manera com interactua amb l'usuari dins d'un flux conversacional estructurat per passos o estats. Aquest enfocament ha permès dissenyar un sistema que, tot i la seva aparent complexitat, es manté modular, comprensible i escalable.

L'anàlisi funcional i tècnica ha servit per establir una base sòlida sobre la qual construir els components clau del sistema. Mitjançant la descomposició del flux en passos identificables, la definició detallada de les seves transicions i condicions, i la construcció d'un model de base de dades estructurat i orientat tant a la configuració com a l'execució, s'ha creat una arquitectura robusta capaç de suportar la gestió d'incidències de manera eficient.

El desenvolupament de la prova de concepte (PoC) valida clarament l'enfocament escollit. I en un futur a través de l'ús de prompts específics i ben definits, la IA pot adaptar el seu comportament a cada cas d'ús, donant respostes contextuais, sol·licitant informació quan caldra o escalant a altres eines com JIRA.

Finalment, l'ús d'eines de test com WCF Test Client, així com la generació de diagrames de flux, seqüència i casos d'ús, ha aportat una solidesa addicional al procés de desenvolupament. Aquestes eines no només han facilitat la validació de cada funcionalitat de forma individual, sinó que també han permès garantir la traçabilitat i coherència del sistema en el seu conjunt. Tot plegat fa que aquest projecte no només sigui funcional en el seu estat actual, sinó també totalment preparat per evolucionar i escalar cap a entorns més complexos i amb major volum de consultes.

5. Bibliografía

[1] OpenAI. (2023). ChatGPT Technical Report.

<https://openai.com/research>

[2] Microsoft. (2024). Documentación de Azure OpenAI Service.

<https://learn.microsoft.com/es-es/azure/cognitive-services/openai>

[3] Atlassian. (2024). JIRA Software Cloud REST API Reference.

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro/>

[4] GCO. (2022). Documentación interna del framework GCO para arquitecturas empresariales. Documentación corporativa interna.

[5] IA Tech - Enlace. (2023). Documentación funcional de los módulos de Analítica de Negocio y Matriz de Categorizaciones. Documentación interna..

[6] IBM. (2022). Architecting Conversational AI: Design principles for virtual agents.

<https://www.ibm.com/cloud/architecture/>

[7] Microsoft. (2024). .NET Core Documentation.

<https://learn.microsoft.com/en-us/dotnet/core/introduction>

[8] International Organization for Standardization. (2011). ISO/IEC 25010:2011 - Systems and software engineering – System and software quality models.

[9] Articles tècnics (2023–2024). Medium, GitHub, Stack Overflow. Consultes puntuals a comunitats de desenvolupament per a resolució de problemes tècnics i bones pràctiques.

